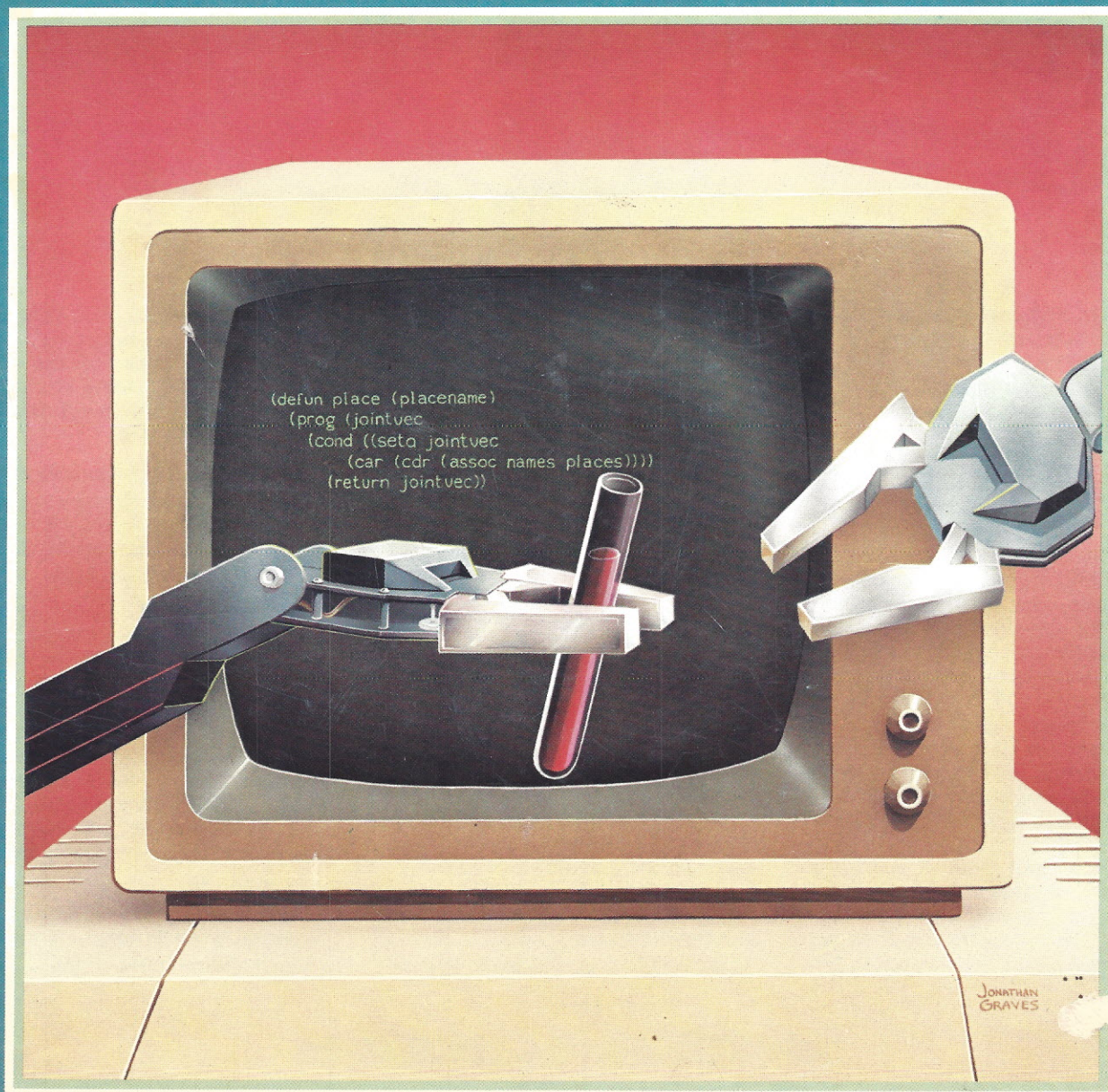
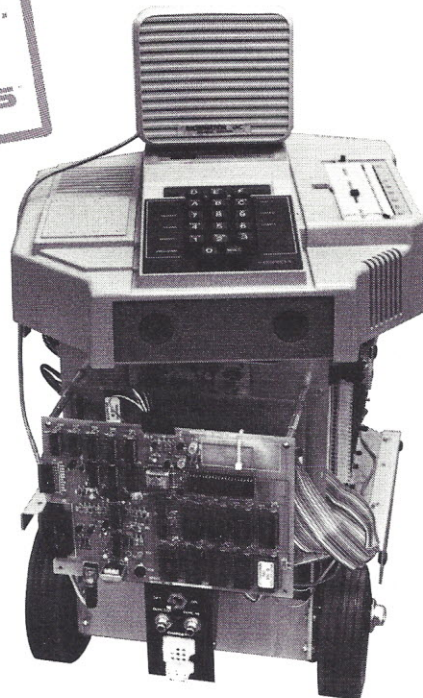


ROBOTICSTM AGE



boost your HERO-1 to powerful new levels



Arctec Systems' whole new line of power-packed peripherals helps your Heathkit HERO-1™ personal robot reach its greatest potential. And with our dynamic software, you can even *talk* to your robot.

Meet Our Peripherals and Software:

HERO MEMCOM™ Board

Develop great, new programs for your HERO-1 robot using your personal computer — and at the same time *boost the robot's memory almost ten-fold with an additional 30K of RAM!*

Our MEMCOM Board comes with an RS-232 serial port for direct communications between your robot and practically *any* computer . . . two 8-bit bi-directional parallel ports for running peripherals — plus two 16-bit timers. Now — make superfast robot-computer data transfers!

You get fast and easy-to-use communications software, for uploading and downloading programs via the serial port . . . you can save programs when the robot is in its "sleep" mode . . .

save programs on disk for later use — and all the hardware and cables needed for easy installation are included. Price: \$345.

Talk to Your Robot—with Our Voice Command System

Connect the versatile **Voice Command System** to the HERO MEMCOM Board, and program your robot's movements *by voice* alone. You get **VOREC** (Voice Recognition Board): a precise, speaker-dependent recognizer, with a flexible vocabulary of up to 256 words you train

to your voice patterns. Plus **VOCOL**: the high-level machine language that gives your robot immediate *and* deferred execution modes. Price: \$595. Voice Command System with HERO MEMCOM Board: \$920.

Apple-HERO Communicator

Comes with all the hardware and software needed to hook up an Apple™ computer to your robot and HERO MEMCOM Board. Price: \$159.

And More

Storyteller . . . Poet . . . Robot Language Macro . . . 6808 Cross Disassembler . . . Text-to-Phoneme Conversion . . . RS-232 Interface . . . Downloader . . . Development Disk for CP/M . . . Robot Cross-Assembler Package . . . Prom Adapter Extender Board . . . Polaroid Rangefinder. Arctec Systems has a full range of exciting add-ons for your HERO-1 personal robot.

Act Right Now

For more detailed information on the incredible range of Arctec add-ons *available today*—write or call.

arctec systems™

9104 Red Branch Road Columbia, Maryland 21045
(301) 730-1237 • Telex 87-781

Apple is the trademark of Apple Computer, Inc.

HERO-1 is a trademark of Heath Corporation.

Publisher/Editorial Director

Carl T. Helmers, Jr.

Editor

Raymond GA Cote

Consulting Editors

Russ Adams

Directory Editor

Heidi Copeland

Production/Design

Nancy Estle

Production Assistant

Tobee Phipps

Circulation Manager

James E. Bingham

Dealer Accounts

Meredith Makela

Circulation Assistant

Trish Harper-Lentricchia

Typography

Maryellen Kelly

Copy Editor

Freida C. Day

Receptionist

Vi Depres

Advertising Manager

Donna Louzier

Advertising Sales

Jerry Merrifield

Judi Wimberly

Robotics Age Inc.

174 Concord Street

Peterborough, NH 03458

603-924-7136

Member of

Business Publications Audit



ROBOTICS AGE—(ISSN 0197-1905) is published monthly by Robotics Age Inc., Strand Building, 174 Concord Street, Peterborough, NH 03458, phone (603) 924-7136. Address subscriptions, changes of address, USPC Form 3579, and fulfillment questions to *Robotics Age* Subscriptions, P.O. Box 358, Peterborough, NH 03458. Second class postage paid at Peterborough, NH and at additional mailing offices. **POSTMASTER:** Send address changes to ROBOTICS AGE, 174 Concord St., Peterborough, NH 03458.

Subscriptions are \$24 for one year (12 issues), \$45 for two years (24 issues), \$63 for three years (36 issues) in the USA and its possessions. In Canada and Mexico, subscriptions are \$28 for one year, \$53 for two years, \$75 for three years. For other countries, subscriptions are \$32 for one year, surface delivery. Air delivery to selected areas at additional charges, rates upon request. Single copy price is \$3 in the U.S., \$3.50 in Canada and Mexico, \$4 in Europe, and \$4.50 elsewhere. Foreign subscriptions and single copy sales should be remitted in United States funds drawn on a U.S. bank.

Address all editorial correspondence to the Editor at *Robotics Age*, Strand Building, 174 Concord Street, Peterborough, NH 03458. Opinions expressed by the authors of articles are not necessarily those of *Robotics Age*. To aid in preparation of acceptable articles, an Authors' Guide is available upon request, if accompanied by a self-addressed 8½ by 11 inch envelope with first class postage for 3 ounces. Unacceptable manuscripts will be returned if accompanied by a self-addressed envelope with sufficient first class postage. Not responsible for lost manuscripts or photos.

Each separate contribution to this issue, and the issue as a collective work, is © 1984 Robotics Age Inc. All rights reserved. Copying done for other than personal or internal reference use without the permission of Robotics Age Inc. is prohibited. Requests for permission should be addressed in writing to *Robotics Age* Permissions, Strand Building, 174 Concord Street, Peterborough, NH 03458.

THE JOURNAL OF INTELLIGENT MACHINES

ROBOTICS AGE™

NOVEMBER 1984

VOL. 6 NO. 11

EDITORIAL

2 Basics First

by Raymond GA Cote

FEATURES

8 A Lisp-Based Robot Control System

by Doug Snead and
John Roach

Part I: Basic Functions

A flexible robot control system can be created with an Apple II computer and a Lisp interpreter.

13 Introduction to Prolog

by Dean A. Schlobohm

Part I: The Commands

A high-level language for logic programming.

20 EARL An Easy Robot Language

by D. S. Holmes

A simple feedback experiment demonstrates the flexibility of modular computer interface devices.

22 Product Notes: The String Computer

by Russ Adams

This chip allows your computer to find near misses.

26 A Biological Model For Multilegged Robot Control

by Thomas A. Easton

Duplicating the reflexes which nature has developed over eons of evolution can provide balance for bipedal or quadrupedal robots.

DEPARTMENTS

3 Sensory Perceptions

6 Calendar

19 Classified Advertising

19 Advertiser Index

34 New Products



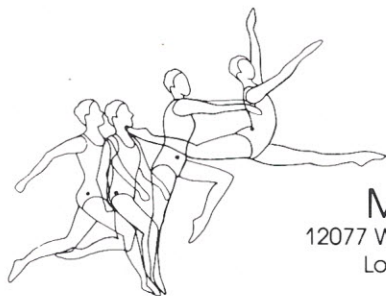
About the cover: Although almost any computer language can be used for robot control, the Lisp language is ideal for describing robot system concepts. This cover by Jonathan Graves depicts a two-armed robot system, controlled by a Lisp program, running on an Apple II computer. Although this particular arrangement is not well suited for real-time control design, it does provide an inexpensive environment for experimenting with such concepts as two-arm control.

MicroMotion

MasterFORTH

It's here — the next generation of MicroMotion Forth.

- Meets all provisions, extensions and experimental proposals of the FORTH-83 International Standard.
- Uses the host operating system file structure (APPLE DOS 3.3 & CP/M 2.x).
- Built-in micro-assembler with numeric local labels.
- A full screen editor is provided which includes 16 x 64 format, can push & pop more than one line, user definable controls, upper/lower case keyboard entry, A COPY utility moves screens within & between lines, line stack, redefinable control keys, and search & replace commands.
- Includes all file primitives described in Kernigan and Plauser's *Software Tools*.
- The input and output streams are fully redirectable.
- The editor, assembler and screen copy utilities are provided as relocatable object modules. They are brought into the dictionary on demand and may be released with a single command.
- Many key nucleus commands are vectored. Error handling, number parsing, keyboard translation and so on can be redefined as needed by user programs. They are automatically returned to their previous definitions when the program is forgotten.
- The string-handling package is the finest and most complete available.
- A listing of the nucleus is provided as part of the documentation.
- The language implementation exactly matches the one described in *MASTERING FORTH*, by Anderson & Tracy. This 200 page tutorial and reference manual is included with MasterFORTH.
- Floating Point & HIRES options available.
- Available for APPLE II/II+//IIe & CP/M 2.x users.
- MasterFORTH — \$100.00. FP & HIRES — \$40.00 each
- Publications
 - *MASTERING FORTH* — \$20.00
 - 83 International Standard — \$15.00
 - FORTH-83 Source Listing 6502.Z-80,8086 — \$20.00 each.



Contact:

MicroMotion

12077 Wilshire Blvd., Ste. 506
Los Angeles, CA 90025
(213) 821-4340

Editorial

BASICS FIRST

I have often heard the current state of the personal robotics industry compared to the Altair days of personal computers. Just like the earliest "hobbyist" computers, today's "personal" robots are usually quite limited in function and difficult to use. These machines do, however, show the first glimmerings of the many possibilities inherent in the small robotic device.

One major difference does exist between today's experimental robot technology and the Altair days of personal computers. The Altair-style computer provided a working base with expansion slots for adding your own carefully constructed circuits or additional commercially-purchased boards. You weren't just handed a box and told "This is as big, powerful, and complex as the computers are going to be." These early computers provided the first building block, the basis for experimentation with data communications, additional memory, and diskette controllers. Eventually, the more complex cards such as light pens, power controllers, motor drivers, and speech synthesizers were created to fit in this same little box. Each machine was an engineering test bed for the next generation computer. In comparison, today's personal robot market has not yet reached the Altair stage. We are still waiting for that first building block.

What should a base robot building block look like? How about a short, squat box (maybe two or three feet tall), on wheels (three or four), that contains an onboard power supply, a processor for which there is plenty of readily available software (such as the Z80), at least 64 Kbytes of programmable memory, a single disk drive, an RS-232 communications port, motor controllers, and most importantly, expansion capabilities.

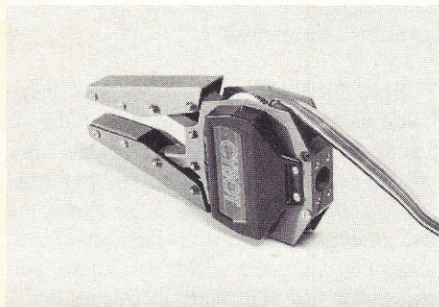
The system must allow electronic, mechanical, and software expansion. An expansion bus (such as the S-100, STD, or Euro-card Bus), would allow additional proprietary and commercial boards to be added to the system. The electronics expansion bus **must not** be proprietary. Haven't we all learned our lesson trying to make different manufacturers' products speak to each other with personal computers? Use a product that is already well defined and supported by many manufacturers.

The base robot system must also support mechanical expansion. Its body could look like an Erector™ set with mounting holes, braces, and brackets for adding scanners, arms, trays, eyes, etc. Don't be overly concerned about how your robot would look running around a household. At this stage of the game, none of these machines are running around the house, they are running around basements, garages, and school laboratories. Worry instead about your system's design flexibility. If your base robot design can be molded and shaped into the form required for experimentation, you've developed a useful base product. More importantly, you've also created a more salable product.

For all the mass media hype, "personal" robots are still a long way from reality. If you, as a designer or manufacturer, are considering what type of robot system to manufacture, you must realize that you can never construct a product that satisfies everyone. But, if you create a flexible, malleable robot, you will have created a product to spark the imagination and perhaps ignite an industry. ■

Sensory Perceptions

CLASSY GRIPPER. The first two impressions I received from the Cybot Gripper™ were quality and class. The quality is evident in the anodized aluminum members and the rugged nylon gears. This industrial-grade gripper is designed for education, research, experimentation, and light industrial applications. It has a 0 in. to 2 in. grip range and a high squeeze-force-to-weight ratio. The gripper jaws are driven by a counter-rotating worm gearing system that is part of a parallelogram link system. The mechanical advantage offered by this geometry lets the jaws exert a 40-pound squeeze force when wide open. This force can be attained with



only 2 V applied to the DC motor. The gripper I am experimenting with has a demo box attached which runs the motor off DC batteries. These batteries have run several hours and still manage to generate a fair bit of gripping force.

Although the Cybot documentation is simply a four-page brochure, it is packed with information and all the data required to connect and operate the gripper. Five methods of operating the DC gripper motor are described, with complete circuitry. Examples of a single power supply linear motor driver without servocontrol, a linear servocontrol motor driver, a simple relay control, a pulse-width modulation control, and a discrete component motor control are provided. Cybot also provides a simple circuit, consisting of a single resistor and two voltage sensors, that can be used to monitor the torque generated by the motor. The gripper jaw position is continuously measured by a potentiometer that travels through 90 degrees. The available output voltage depends on the input voltage to the potentiometer. Usually, this is derived from a system's +5 V or +15 V supply. The sensor's output produces 0 V at the *jaws*

closed position and 1/3 of the input voltage (+5 V or +15 V) at the *jaws open* position. Conveniently, the potentiometer output is linear to the jaw angle.

Two methods are provided for mounting the gripper. A universal mount with four through-tapped holes makes connection to sheet metal arms easy. Similarly, a clamp is provided to attach the gripper to a 0.5 in. shaft. An 18 in. cable and connector is supplied to provide power to the DC motor and outputs for sensing gripper position and force.

Having seen and used this gripper for several days, I eagerly look forward to the next product from Cybot—their Cybot Tutor robot. Pictures indicate this educational robot arm is constructed from the same anodized aluminum material. I congratulate Cybot for bringing their high-caliber, high-quality, industrial-grade products into the experimental and educational marketplace.

SECURITY ASSOCIATION. TWS Security is forming an association for security professionals, software designers, technicians,

Tabletop Technology™

An Inventor's Toolchest for Ideation & Experimentation

Now you can have, at your fingertips, a comprehensive collection of innovative engineering tools. At a truly reasonable price.

Crystalize your ideas by constructing model mechanical, electro-mechanical, pneumatic, and electronic devices in minutes. Simply by snapping together *functioning* components.

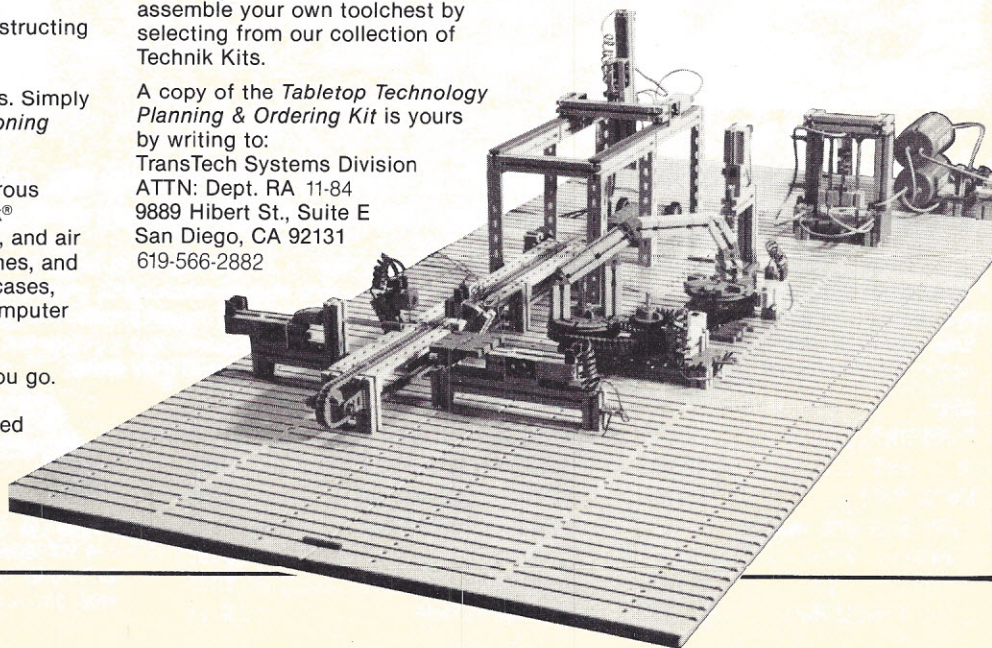
The system includes a generous assortment of fischertechnik® girders, gears, cams, motors, and air cylinders. Plus valves, switches, and integrated circuits. Storage cases, benches, baseplates, and computer interfaces too!

Simple enough to learn as you go. Yet sophisticated enough to simulate a computer-integrated manufacturing plant—complete with conveyors, robotic cells, and machining centers.

The Tabletop Technology Mini-Lab™ provides an instant R & D modeling station, supplied with all system components. You can also custom-assemble your own toolchest by selecting from our collection of Technik Kits.

A copy of the *Tabletop Technology Planning & Ordering Kit* is yours by writing to:
TransTech Systems Division
ATTN: Dept. RA 11-84
9889 Hibert St., Suite E
San Diego, CA 92131
619-566-2882

FROM
CREATIVE LEARNING
SYSTEMS, INC.



Sensory Perceptions

amateurs, etc. who are interested in security applications for robotics, Computer Aided Design, data security, personal computers, or other areas of high technology security.

The society will act as a clearinghouse for information, designs, and research. It will also seek to represent this new area of security with local, national, and international security associations. The society will strive for greater recognition of the possibilities of this new area of security and increased sales of these types of products. TWS Security believes that computers and robotics could alter the entire security field by the late 1990s, eliminating needless loss of human life, providing simulations to fight terrorism, saving millions of dollars in credit theft, and preventing the loss of valuable electronic data.

TWS Security currently markets a queueing simulation designed to combat retail crime by more effective use of security personnel. They have also built a prototype electronic watchdog, MAX, who can detect fires, patrol a home in a set pattern, simulate the occupants of the home to discourage burglars, and sound an

alarm if the home is invaded by unwanted persons. Plans call for MAX II to add small fire-fighting tools, more advanced sensors, and possibly remote TV broadcast.

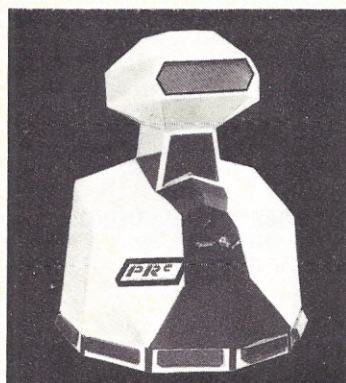
Anyone interested in exchanging information or joining such an organization should contact Lester S. Rayburn, The Robotics Security Society, 17 Watts St, Kittery, ME 03904.

THE ROBOT I BUILT DURING MY SUMMER VACATION. How about a summer camp for robotics? Last summer, the Science Museum of West Palm Beach, Florida held a week-long program wherein students learned the fundamentals of robotics. The object was not just to use robots for particular projects, but to actually construct them. Q-Bot kits, loaned to the museum by Eastern Machinery and Manufacturing, Inc. of Salt Lake City, contained the computer-robot interface electronics and a wide variety of motors, sensors, and structural components. Four types of robots were assigned as projects: one painted pictures, one washed small windows, one squeezed orange juice, and one (named GRETA) located tin cans, picked them

up, and dropped them into an attached trash bin. By the end of camp, all the robots were busily, if not always reliably, performing their appointed tasks.

According to Edwin J. C. Sobey, Ph.D., Executive Director of the Science Museum, "Robotics camps are inherently more stimulating than computer camps. The students do still need to study computer programming. But they use the computer to control the robots in addition to processing information. They can also experience the thrill of designing and building their own machines and making them work. The physical work and the satisfaction of seeing and showing off their products are rewards in themselves."

PUBLIC ODETTICS. Odettics, Inc. of Anaheim, California, the company that brought us the six-legged walking robot, presented a public stock offering on August 9. Dr. Isaac Asimov, credited as being the person responsible for coining the term *robotics*, and Chairman Joel Slutzky were present for the opening bids. Dr. Asimov purchased the first 100 shares of Odettics, Inc.

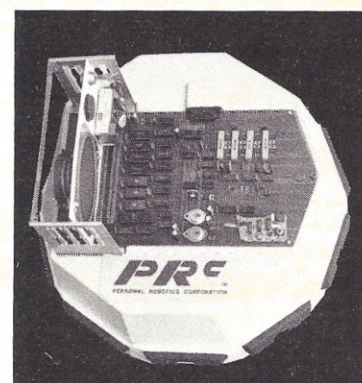


RoPet-HR™

Self-Contained Personal Robots

RoPet™

"1st Place-Commercial Division"
2nd Tournament of Robots, April 29, 1984
"Best Overall-Commercial Division"
Tournament of Robots, October 16, 1983



RoPet-XR™

Whether kit or fully-assembled, RoPet™ is designed to be easy-to-use, reliable, expandable,...and USEFUL: RoPet™ is your Robot Guard-Dog, requiring spoken password or it sounds the alarm!

RoPet's Powerful Features: Z80 Computer, 4 blank S-100 bus slots, 8K 6116 RAM, 8K 2716 EPROM, ZIF Socket for Driver EPROM, 3 Driver EPROMs provided (HomeSentry™, VerbaLogo™, SpeechTrain™), up to 256 word vocabulary "Hands-Free" speech recognition (typ. 95+% accuracy) with battery backup/msec response time, complex-sound and infinite vocabulary speech generation (w/10-Watt speaker), color organ, key-lock security switch, long-lasting (typ. 8 hr) rechargeable batteries, low-voltage detection, battery charger, sonar, and passive infra-red detection. All kits are guaranteed-upgradable to the RoPet-XR™ Personal Robot. Manual each kit: \$25 is deductible from purchase price of kit. CA residents add 6½% sales tax. Send \$1.00 for catalog.

A&T:

RoPet-HR™	\$2,495
RoPet-XR™	\$2,195
Motherboard	\$1,295
RoPet Base (inc. motors, batteries, wheel encoders, etc.)	\$ 595
S-100 EPROM Burner	\$ 245

Kits:

Motherboard (bare) for kits	\$145
Speech Recognition kit	\$295
Speech/Complex Sound kit	\$195
Computer/S-100 Bus kit	\$295
Sonar kit	\$195
Motor-Controller kit	\$145
Infra-Red Module	\$ 95



PERSONAL ROBOTICS CORPORATION
469 WASKOW DRIVE
SAN JOSE, CA 95123
(408) 281-7648

Shipping/handling: \$45 ea. RoPet™ else \$10

Dealer Inquiries Invited

Visa • MasterCard • Check • Money Order



Dr. Dobb's Stands Apart

Take a good, hard look at the crowded computer magazine market. If you're a serious microcomputerist, one magazine stands apart. **Dr. Dobb's Journal**.

Dr. Dobb's is not for everyone. It is written by and for expert programmers, and it's the oldest and most technically sophisticated microcomputer publication available. Since 1976, **Dr. Dobb's Journal** has been the unchallenged leading source of software tools for advanced programmers.

With the industry moving forward faster than ever, you need to stay a step ahead. **Dr. Dobb's** sets the pace with:

- more actual code than *Microsystems*, *Microcomputing* or *Byte*;
- regular columns on C, Unix, CP/M and 16-bit software;
- algorithms and problem solving;
- lively discussions of fundamental software issues;
- inside information on commercial languages and operating systems;
- tips on advanced programming topics.

The information and valuable code contained in **Dr. Dobb's** makes each issue indispensable for serious computing professionals and enthusiasts. Don't miss a single issue of this valuable resource. Subscribe today. If you aren't fully satisfied, cancel your subscription and keep the first issue as a free sample.

To start your subscription, fill out this coupon and return it to:

Dr. Dobb's Journal
P.O. Box 27809, San Diego, CA 92128

Yes! Please enter my subscription for 12 issues of:
Dr. Dobb's Journal for \$25.

If I am not fully satisfied I will write "cancel" on my subscription invoice and keep the first issue as a free sample.

Name _____

Address _____

City _____

State _____ Zip _____

☐ Bill me later ☐ I've enclosed \$25 Charge my ☐ VISA ☐ M/C

Card No. _____

Expiration _____

Signature _____

Offer good in U.S. only and expires Feb. 28, 1985

4002

SOFTWARE TOOLS FOR ADVANCED PROGRAMMERS
Dr. Dobb's Journal

Calendar

November 1984

4-9 November. Intelligent Robots and Computer Vision. Hyatt Regency, Cambridge, MA. Contact: SPIE—The International Society for Optical Engineering, Bellingham, WA 98227-0010, telephone (206) 676-3290.

The emphasis for the 1984 conference is on intelligent robots, vision, and sensor systems. The intent of the sessions is to address and interchange ideas on a wide range of applications, issues, and techniques for these advanced intelligent systems. State-of-the-art technical design and modern equipment concepts will be presented. Topics to be addressed and for which contributed papers are requested include: government and industrial programs in intelligent sensors and robotics; applications of intelligent sensor systems; statics, kinematics, dynamics, and control of robot systems; standards for sensors and robots; image processing; feature extraction; three-dimensional object representation; use and application of artificial intelligence; and image understanding in intelligent robotics and computer vision.

9-10 November. 23rd Annual Illinois Association of Electricity, Electronic Educators Convention/Workshop—1984. Holiday Inn, East Peoria, IL 61635, telephone (309) 694-5495.

This year the convention workshops will focus on a wide variety of subjects. Tentative topics include industrial electronics, microprocessors, and digital electronics. Industrial and educational representatives will be there with their latest equipment and products.

13-14 November. Intelligent Robots: Achievements and Issues. SRI International, Menlo Park, CA. Contact: Cherry Powers, SRI International, Room EL219, 333 Ravenswood Ave., Menlo Park, CA 94025.

Under a grant from the National Science Foundation, SRI International will conduct a two-day workshop co-chaired by Dr. David Nitzan, Director, Robotics Laboratory, and Dr. Robert Bolles, Senior Computer Scientist. Up to 50 workshop participants will be invited to discuss significant achievements and issues in seven principal areas of robotics research:

mobility, manipulation, noncontact sensing, contact sensing, adaptive control, off-line programming and planning, and robotic CAD/CAM system integration. In addition, up to 150 people from academic, industrial, and government organizations will be able to attend the workshop as observers and participate in two open-discussion sessions.

13-15 November. Man or Machine—A Choice of Intelligence. Sheraton Sand Key Hotel, Clearwater Beach, FL. Contact: Rhonda Gerganess, CAM-I, 611 Ryan Plaza Drive, Suite 1107, Arlington, TX 76011.

Decision processing methods and their application will be the theme of CAM-I's fall conference. The conference will address a number of issues that have developed during CAM-I research projects. In particular, the issue of what type of decision process should be used for particular projects will be discussed. Indications are that there is no one system that meets all requirements. Some situations are best served by Decision Trees while others by Expert Systems or true Artificial Intelligence methods. Although the conference will attempt to make distinctions between the different methods, the purpose is not to create rigid definitions of decision processing methods.

16-17 November. 6th Annual Forth Convention. Hyatt Palo Alto, Palo Alto, CA. Contact: Forth Interest Group, PO Box 1105, San Carlos, CA 94070, telephone (415) 962-8653.

The Forth Interest Group is a worldwide non-profit organization of over 4700 members and 53 chapters devoted to the Forth computer language. The convention is prepared to meet the needs of every Forth enthusiast—beginner to professional—with two days of hands-on tutorials, exhibits/vendor booths, lectures, and discussions.

23-25 November. Forth Modification Laboratory (FORML). Asilomar, Pacific Grove, CA. Contact: Forth Interest Group, PO Box 1105, San Carlos, CA 94070, telephone (415) 962-8653.

FORML conferences provide a forum for sharing and discussing new proposals intended to benefit the Forth computer language. Expert systems and artificial intelligence are among the many topics that will be covered.

27-30 November. Robots-West. Anaheim Convention Center, Anaheim, CA. Contact: Jeff Burnstein, Robot Industries Association, PO Box 1366, Dearborn, MI 48121, telephone (313) 271-0778.

TOTAL CONTROL:

FORTH: FOR Z-80®, 8086, 68000, and IBM® PC

Complies with the New 83-Standard

**GRAPHICS • GAMES • COMMUNICATIONS • ROBOTICS
DATA ACQUISITION • PROCESS CONTROL**

- **FORTH** programs are instantly portable across the four most popular microprocessors.

- **FORTH** is interactive and conversational, but 20 times faster than BASIC.

- **FORTH** programs are highly structured, modular, easy to maintain.

- **FORTH** affords direct control over all interrupts, memory locations, and i/o ports.

- **FORTH** allows full access to DOS files and functions.

- **FORTH** application programs can be compiled into turnkey COM files and distributed with no license fee.

- **FORTH** Cross Compilers are available for ROM'ed or disk based applications on most microprocessors.

Trademarks: IBM, International Business Machines Corp.; CP/M, Digital Research Inc.; PC/Forth and PC/GEN, Laboratory Microsystems, Inc.

FORTH Application Development Systems include interpreter/compiler with virtual memory management and multi-tasking, assembler, full screen editor, decompiler, utilities and 200 page manual. Standard random access files used for screen storage, extensions provided for access to all operating system functions.

Z-80 FORTH for CP/M® 2.2 or MP/M II, \$100.00;

8080 FORTH for CP/M 2.2 or MP/M II, \$100.00;

8086 FORTH for CP/M-86 or MS-DOS, \$100.00;

PC/FORTH for PC-DOS, CP/M-86, or CCPM, \$100.00; **68000 FORTH** for CP/M-68K, \$250.00.

FORTH + Systems are 32 bit implementations that allow creation of programs as large as 1 megabyte. The entire memory address space of the 68000 or 8086/88 is supported directly.

PC FORTH + \$250.00

8086 FORTH + for CP/M-86 or MS-DOS \$250.00

68000 FORTH + for CP/M-68K \$400.00

Extension Packages available include: software floating point, cross compilers, INTEL 8087 support, AMD 9511 support, advanced color graphics, custom character sets, symbolic debugger, telecommunications, cross reference utility, B-tree file manager. Write for brochure.



Laboratory Microsystems Incorporated
Post Office Box 10430, Marina del Rey, CA 90295
Phone credit card orders to (213) 306-7412



Calendar

Robots-West is sponsored jointly by Robotic Industries Association (RIA), and Robotics International of the Society of Manufacturing Engineers (RI/SME). It will feature exhibits by leading robot manufacturers and component suppliers. Approximately 6,000 visitors are expected to attend the three-day exposition and conference. Individual papers will discuss robots in aerospace, nuclear, construction, agriculture, and harsh environments. Other topics include displaced worker retraining, labor's response to automation, robots in mining, software/programming languages, and sensors.

December 1984

3-4 December. **Expert Systems: A Practical Application of Artificial Intelligence (Course #1120DC)**. George Washington University, Washington, DC 20052. Contact: J. Perkins, George Washington University, Washington, DC 20052, telephone (202) 676-8510.

A continuing education course designed to enhance and expand the technical skills of practicing professionals in the field of Artificial Intelligence. This course identifies the many areas in which expert systems are being applied and presents the fundamental concepts and components of systems. It presents the various design issues and discusses methods for testing, validation, and evaluation of systems. Implementation, management issues, and future trends are also examined.

February 1985

26-28 February. **Agri-Mation 1**. Palmer House Hotel, Chicago, IL. Contact: SME Public Relations, One SME Drive, PO Box 930, Dearborn, MI 48121, telephone (313) 271-1500.

The three-day exposition will feature demonstrations of a wide range of automated equipment used in the production of food, fiber, and biomass materials. Featured will be new developments in irrigation and drainage, field equipment, material handling, storage and distribution, harvesting, processing, and the latest in computerized systems and services available to the agricultural manager, researcher, and engineer. Among the agricultural automation products on display will be data acquisition systems, measurement instruments, mobile robots, motors, programmable controllers, robotic components and systems, sensors, controls, transducers, and actuators.

March 1985

26-28 March. **Vision '85**. Cobo Hall, Detroit, MI. Contact: Jeff Burnstein, Robotic Industries

Association, PO Box 1366, Dearborn, MI 48121, telephone (313) 271-7800.

Vision '85 will be the first major exhibition of machine vision systems and related equipment ever held. A comprehensive technical conference will accompany the show. RIA's sponsorship of Vision '85 is an outgrowth of the trade association's new emphasis on machine vision.

April 1985

22-25 April. **LASERBOTICS: Combining Laser and Robot Technologies**. Ann Arbor, MI. Contact: Steve Palma, SME Special Programs Department, One SME Drive, PO Box 930, Dearborn, MI 48121, telephone (313) 271-1500.

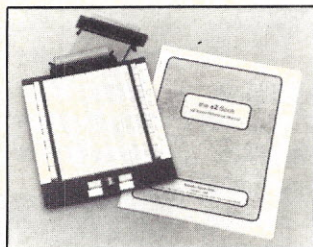
The program will feature speakers from the U.S., Europe, and Asia and will examine the latest advancements in the science of combining lasers and robots to improve productivity. Co-chairperson Jack Lane, director of the Robotics Center at the GMI Engineering and Management Institute, and David Belforte, President, Belforte Associates, are preparing an

agenda covering the latest advancements in laser tooling, robotic part presentation, laser guided robotics, fiber optics, laser/robot welding and inspection.

23-24 April. **1985 Conference on Intelligent Systems and Machines**. Oakland University, Rochester, MI. Contact: Professor Nan K. Loh, Conference Chairman, Center for Robotics and Advanced Automation, School of Engineering and Computer Science, Oakland University, Rochester, MI 48063.

Technical papers are requested which reflect both advances and applications in all aspects of intelligent systems and machines. Suggested topics include: intelligent robots, machine intelligence, adaptive control and estimation, visual perception, artificial intelligence for engineering design, intelligent simulation tools, computer-integrated manufacturing systems, knowledge representation, expert systems, game theory and military strategy, interpretation of multisensor information, and automatic programming. Authors are requested to submit a 300- to 500-word abstract by 1 December 1984.

BUILD YOUR OWN UNIQUE INTERFACES QUICKLY AND EASILY



Pat. Pending

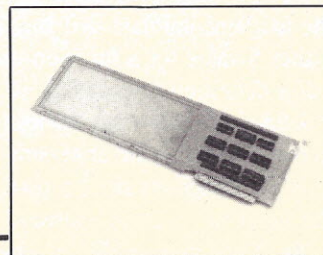
eZ Card — the low cost plug-in prototyping board for IBM-PC and hardware compatible computers. Features include a buffered address, data and control bus and switch selectable address decoder. A large prototype development area consisting of over 2,300 holes on a 0.1" grid easily accommodates over 60 IC sockets for wire-wrapping your own specialized PC interfaces. Gold plated edge connector ensures maximum reliability. **eZ Card** comes complete with the **eZ Book** and mounting bracket for installation into the system unit.

ONLY \$89.95 plus \$5.00 S&H*

The convenient, fast, low-cost way to build your own developmental circuits or test fixtures. **eZ Board** plugs into your computer's expansion slot through an integral 18-inch ribbon cable. An array of terminal blocks provide access to each of the signals made available from the computer bus. Signal and power identification (exactly the same as used by the computer manufacturer) is clearly and permanently marked adjacent to each pair of terminals. Large solderless breadboard lets you plug in DIPs plus discrete devices and simply interconnect with ordinary hookup wire. No solder, no wire-wrapping needed. System comes complete with **eZ Book** which contains several practical and useful projects such as a/d converter, parallel port, joystick interface, etc. and software listings.

IPC64 **eZ Board** for IBM-PC
APC50 **eZ Board** for Apple II & IIe
CPC44 **eZ Board** for Commodore 64

ONLY \$124.95 plus \$5.00 S&H*



*USA Only. Overseas add \$20.00 S&H (Airmail). California residents add 6% Sales Tax.

Send check or money order to:

SABADIA EXPORT CORPORATION
BOX 1132, YORBA LINDA, CA 92686
(714) 630-9335 / Telex 756-582 Easylink.
Dealer/Distributor inquiries welcome.

A LISP-BASED ROBOT CONTROL SYSTEM

Part I: Basic Functions

Doug Snead
John Roach

Department of Computer Science
Virginia Polytechnic Institute
Blacksburg, VA 24061

Choosing the appropriate language for a robot control system is very important. Anything the robot must do, know, or sense must have its logic expressed in the chosen language. A language should be chosen that is powerful, easy to use and understand, and has available software. If it is for an experimentalist, it must also be reasonably cheap. Assembly language (or worse, machine code) is inexpensive, but is difficult and time-consuming to produce, modify, and understand. BASIC is widely available for both small and large computers, is easy to learn, and interactive. However, many useful data structures, such as trees, can only be implemented with great difficulty. A more serious problem with BASIC programs is that they tend to become nonmodular since all variables are global and the variable names are often restricted to a single, nonmnemonic letter or letter and digit combination. Nonmodularity is also increased by the necessity to branch (to a nonmnemonic address) to alter program flow.

Existing Artificial Intelligence (AI) software programs, reported in journal articles and technical reports, are never written in BASIC, but are often expressed in Lisp. Lisp was invented in the late 1950s specifically as a symbol manipulation language for AI research.

Lisp is both powerful and easy to use; powerful in the sense that very high-level algorithms can be expressed concisely, and easy to use because Lisp is (primarily) an interpreted language. The Lisp interpretation environment allows you to interrupt a running program at any time, examine variable values, perform program tracing, examine the run-time stack, and modify other interpreted functions. Lisp programs are executed by *evaluating* them. The *evaluation* routine is itself a function called *eval*.

The fundamental data structure in Lisp is the list. A list may be either a value (of a function or variable, for example) or a function definition, depending on the context. Ultimately, the user-defined functions are themselves run using a list containing a function name and arguments. This uniformity of data and code is advantageous since a useful modification or extension to the language is straightforward in Lisp, whereas it may be difficult, if not impossible, in other languages.

Another advantage to using Lisp is the ability to form other languages. A common Lisp practice is to define some special-purpose language that shares the basic data structure of Lisp—the list. The new language is then implemented in Lisp. A well-known example is Terry Winograd's SHRDLU natural language

understanding program. Imbedded in this Lisp program is a set of Lisp functions to interpret the theorem-proving language Planner. Another set of Lisp functions implemented the parsing language Programmer. Both Planner and Programmer could be viewed either as separate, special-purpose languages, each with a unique syntax and semantics, or they could be viewed as sets of subfunctions in the context of the whole SHRDLU program.

Although Lisp was designed as a research language, several implementations exist that have been used to produce commercial products. MACSYMA is a large set of Lisp programs that allow you to solve mathematical problems symbolically using algebraic rules and transforms. You can expand and simplify transforms, find derivatives and integrals, and perform Laplace transforms. In organic chemistry, DENDRAL is a well-known Lisp system that can predict the molecular structure of an unknown molecule given its mass spectrogram data. In medicine, MYCIN is a Lisp-based expert consultant system that uses the rules of medical experts to decide on the appropriate antibiotic treatment for various bacterial infections. The list goes on and on, because in the 20 years that Lisp has been around, many AI programs have been expressed in Lisp.

Above all, Lisp is easy to use. Although this may seem contradictory—a research language that is easy to use—the uniformity of program and data helps simplify design and debug-

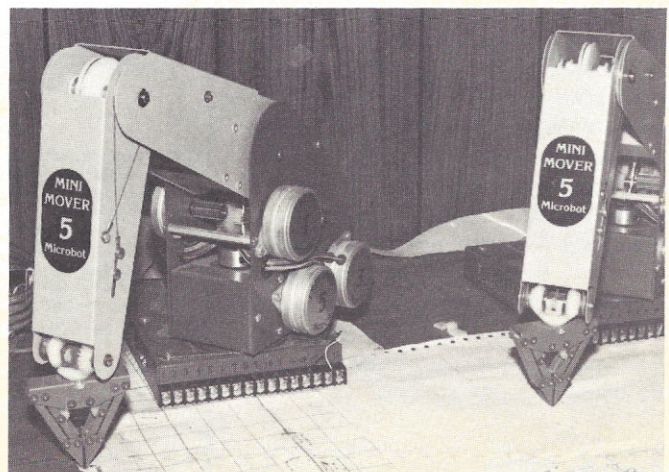


Photo 1. The Lisp-based robot control system can be used to control up to two robot arms. The specific system used here controls two Microbot MiniMover-5 table-top robot arms.

ging. Programs tend to be more understandable since functions tend to be small. Thus, the problem at hand is naturally divided into logical modules.

Usable, affordable Lisp systems are also available for many popular microcomputers. The Lisp used in this article runs on the Apple II™ and costs less than \$130. Combined with the MiniMover-5™ robot arm, this system is available off the shelf and offers an attractive route for those interested in robotics, but who don't want to bother with building the required hardware.

This article describes a robot control system written in Lisp on the Apple II computer. The system is designed to be as general as possible and allow a wide variety of tasks to be implemented with the defined tasks. The stated goal of the robot control system is to perform several primitive actions such as remembering arm positions by name, and keeping track of the positions of various objects in the robot's environment. Ultimately, you can define a series of robot commands built on these basic functions that form an extension to the basic Lisp system by adding grasp, release, and move operators. For the sake of simplicity, the system does not deal with XYZ coordinates, but instead refers to *places* as a vector (a list of integers, six long in this case) of joint positions that the arm must have in order to be at that place. The generation of a joint vector given an XYZ coordinate is useful, but is a complex mathematical problem and won't be discussed here. Once the XYZ conversion routine is written, however, it would fit well within the framework of the system described in this article.

LOW-LEVEL CONTROL

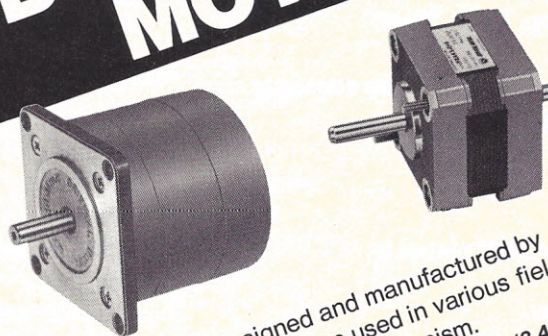
At the lowest level of arm control are the routines to control the physical arm movement. For the Apple II and MiniMover-5, a series of BASIC firmware commands are provided. These include a step command (to move each of the joints some number of steps forward or backward), a close command (to close the gripper until an object is gripped), a set command (to allow you to move any joint of the arm by pressing keys on the Apple II keyboard), and a read command (to return the current recorded position of the arm). These functions are meant to be invoked in BASIC and need intermediate functions in Lisp to do the character padding so they may be used in Lisp. The interface routines are specific to the Apple II—MiniMover-5 system, but other robot arms could be used with the system described here by coding these low-level routines for any particular arm.

CONTROL FUNCTIONS

The MiniMover-5 communicates to the Apple II via a control card attached to the Apple II internal bus. The card contains interface logic and the control programs in EPROM. In BASIC, it is necessary to execute the `PR#n` command to redirect output to a particular Apple II slot. The equivalent command in Lisp is `(pr#n)`. Once the `(pr#n)` command is executed, all output that would normally go to the screen will now go to the device in slot `n`. The robot control system uses a global variable `*armslot` that is initialized to the slot number of the MiniMover-5 robot control card.

Commands are transmitted to the MiniMover-5 through a string containing a set of instructions. For example, sending the character string `@step;n1;n2;n3;n4;n5;n6;d` to the controller card

From 2-Phase to 5-Phase DC STEPPING MOTORS



DC stepping motors designed and manufactured by "ORIENTAL MOTOR" have been used in various fields to control speed, position and synchronism.

2-Phase Hybrid Type: 0.9°/1.8° Mounting Size: 2.3"sq/3.4"sq
0.9°/1.8°/3.6° Mounting Size: 1.7"sq



ORIENTAL MOTOR®

Tokyo, Japan

ORIENTAL MOTOR U.S.A. CORP.

Los Angeles 213/515-2264 New York 201/882-0480
Chicago 312/577-0310 Boston 617/568-8514
San Jose 408/988-2655

Visit our booth at
WESCON

Now Your Computer Can See! \$295.00*

A total imaging system complete and ready for plug-and-go operation with your personal computer.

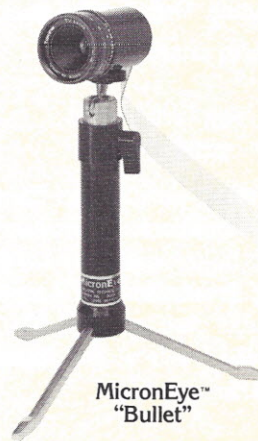
The MicronEye™ offers selectable resolution modes of 256 x 128 and 128 x 64 with operating speeds up to 15 FPS. An electronic shutter is easily controlled by software or manual functions, and the included sample programs allow you to continuously scan, freeze frame, frame store, frame compare, print and produce pictures in shades of grey from the moment you begin operation.

Only the MicronEye™ uses the revolutionary IS32 OpticRAM™ image sensor for automatic solid state image digitizing, with capability for grey-tone imaging through multiple scans. And with these features, the MicronEye™ is perfectly suited for graphics input, robotics, text and pattern recognition, security, digitizing, automated process control and many other applications.

The MicronEye™ is available with immediate delivery for these computers: Apple II, IBM PC, Commodore 64 and the TRS-80CC (trademarks of Apple Computer Inc., International Business Machines, Commodore Corp., and Tandy Corp. respectively).

Phone for MicronEye™ information on the Macintosh, TI PC and RS232 (trademarks of Apple Computer Inc. and Texas Instruments respectively.)

*(Add \$10.00 for shipping and handling [Federal Express Standard Air]; residents of the following states must add sales tax: AK, AZ, CA, CO, CT, FL, GA, IA, ID, IL, IN, LA, MA, MD, ME, MI, MN, NC, NE, NJ, NY, OH, PA, SC, TN, TX, UT, VA, VT, WA, WI.)



MicronEye™
"Bullet"

MICRON TECHNOLOGY, INC.

VISION SYSTEMS
2805 East Columbia Road
Boise, Idaho 83706
(208) 383-4106
TWX 910-970-5973

causes joint j_i to be stepped n_i steps with a delay of d between motor steps. In Lisp, we will use a function (steparm x) where x is a list of steps to be applied to each joint of the arm. The delay value is contained in a global variable *delay so that it can be easily changed from anywhere in the program.

The (steparm x) function is defined as follows:

```
(defun steparm (steplist)
  ;step the arm
  ;joints
  (progn (prin1 '@step')
    ;print '@step',
    ;no <CR>
    no <CR>
    ;steparm
    ;verification
    ;now output
    ;steplist
    ;print elements of
    ;steplist
    (defun prlist (steplist)
      ;print elements of
      ;steplist
      (cond
        ;with '','' between
        ;each
        ((eq steplist nil) (terpri))
        ;empty list, out-
        ;put <CR>
        (t (prin1 ';')
          (prin1 (car steplist))
          (prlist (cdr steplist))))
      ;output ;element
      ;recur on rest of
      ;line
```

The control card also has a @set command for manually positioning the arm from the Apple II keyboard. You can use the number keys one through six to step the joints one through six forward. The letter keys Q-W-E-R-T-Y step the joints backwards. The space bar stops all motors and the zero key exits the @set routine. Defining the @set command in Lisp is very simple:

```
(defun setarm ()
  ;set joints from
  ;keyboard

  (progn (prin1 '@set,')
    (print *delay)))
```

When using the set function, you need to know the current joint positions. The built-in robot controller routines provide for this by keeping a count of the steps applied to each joint. When the @read command is issued, the routines return the joint position of each arm joint. The robot control system has a function called (readpos) that returns a list of joint positions.

```
(defun readpos ()
  ;return list of
  ;current joint
  ;positions

  (progn (prin1 '@read')
    (list (chrnum (read))
      ;joint pos 1
      (chrnum (read))
      ;joint pos 2
      (chrnum (read))
      ;joint pos 3
      (chrnum (read))
      ;joint pos 4
      (chrnum (read))
      ;joint pos 5
      (chrnum (read))))
    ;joint pos 6
```

The (chrnum x) function changes a sequence of digits into a number. The robot controller card returns leading zeros and Lisp does not convert it into a number but leaves it as a sequence of characters.

```
(defun chrnum (n)
  ;return integer
  ;value of n
```

```
(prog (clist x)
  (cond ((numberp n) (return n)))
  ;it's already an
  ;integer

  (setq clist (explode n))
  ;make a list of
  ;chrs in n

  (setq x 0)
  ;init output value
  ;x

  (setq clist (cdr clist))
  ;remove leading
  ;blank

  loop:
    (setq x (* x 10))
    ;shift ans left
    ;one dec place

    (setq x (+ x (car clist)))
    ;add digit

    (setq clist (cdr clist))
    ;remove first
    ;element

    (cond ((null clist) (return x))
      ;finished, return
      ;ans

      (t (go loop:))))
  ;loop until done
```

Thus (readpos) returns the current arm position as a joint vector of integers.

Since the readpos function is somewhat slow, the Lisp robot control system maintains the arm position as a global variable available throughout the program. Usually, the current location is a byproduct of performing an operation that changes the arm position. Therefore, *here need only be reassigned when the arm's position changes. The only functions that actually move the arm are steparm and setarm. In the final robot control system, steparm is never directly called by a user program. It is used by higher-level functions such as goarm which is defined later. The function goarm takes care of updating *here when the arm is moved.

The previously defined function setarm must be modified to automatically maintain the global variable *here. The modification measures the relative change in position during the actual arm movement with the readpos function and then adds this relative change to the old current arm position stored in *here. The modified setarm function is defined as:

```
(defun setarm ()
  ;user positions
  ;robot arm

  (prog (before)
    (setq before (readpos))
    ;get internal arm
    ;pos

    (prin1 '@set')
    (print *delay)
    (setq *here)
    ;vector addition
    ;def'd later

    (subvec (readpos) before))))
```

The @reset command is provided to initialize the internal counter. All counts are relative to a known starting or home position. The experimentalist must put the arm in the home position and use the @reset command to initialize the internal count to zero.

```
(defun resetarm ()
  ;clear internal
  ;counter *here

  (progn (print '@reset')
    (setq *here '(0 0 0 0 0 0))))
```


The function `resetarm` zeros the internal joint position counts and initializes the current arm position joint vector `*here`.

The software provided with the MiniMover-5 provides the `@close` command to close the gripper around an object. The `@close` Microbot close command closes the gripper one step and checks an I/O port until a certain bit in the I/O port is set, indicating the gripper has stopped closing. The problem with this is you need to look at the port to determine if the gripper is holding anything. The Lisp robot control system does not use the `@close` command. Rather, it uses the `steparm` function already defined. However, there is still the problem of determining how the gripper port can be addressed. The function `port` is defined to return information from a memory-mapped I/O port.

```
(defun port (arm slot port1-7)
  (peek (+ -16264
            (+ (* arm 8)
              (+ (* slot 16)
                port1-7))))))
;Apple I/O port
```

The `port` function returns an integer representing the status of the robot's I/O port. The port that is connected to the closed gripper switch is port 7. If the `(port <arm#> <slot#> 7)` function returns a number greater than 128, then nothing has been gripped. If a number less than or equal to 128 is returned then something is in the gripper. A function `grip` can be defined to return the gripper status.

```
(defun grip ()
  (cond ((> (port 1 *armslot 7) 128)
    'empty)
    (t 'holding)))
```

As mentioned earlier, the robot control routines communicate to the Microbot control card by issuing the `(pr# *armslot)` command, printing commands to the card, and finally using the `(pr# 0)` command to allow printing to appear on the screen. The function `robot-rep` reads a robot command from the keyboard, uses the `(pr# *armslot)` sequence to execute the command, and then asks for another command. These commands are just the Lisp robot functions that are defined throughout this article.

An initialization and exit command should be used at the beginning and end of each session. The `'init` command initializes the robot arm and sets the internal counters to zero. The `'exit` command allows a graceful method of leaving the `robot-rep` function loop. The `robot-rep` function is defined as:

```
(defun robot-rep ()
  (prog (reval)
    loop:
      (print '(enter robot command))
      (setq reval (read))
      (pr# *armslot)
      (setq reval (eval reval))
      (pr# 0)
      (print reval)

      (cond ((eq reval 'exit)
        (return 'robot-rep))
            ((eq reval 'init) (arminit)))
      (go loop:)))
;get command
;eval it
;print return value
```

The following convention should be observed when using `robot-rep`: All printing done is intended to be command strings

WHAT'S DIFFERENT ABOUT THIS ROBOTICS AD FROM ALL THE OTHERS?



For a close-up look at the Robotics Educational System, visit our booth #230-228 at the AVA Convention in Louisiana, December 1-3. Also, call about our Fall Robotics Workshop Schedule.

Call PREP, Inc. 800-257-5234; 609-882-2668
or write: Marketing, 1007 Whitehead Road
Ext., Trenton, NJ 08638.

Answers:
1. Hardware
2. CURRICULUM
3. Software

for the robot control card. The `msg` function should be used to print any messages on the screen from within a robot-rep command.

```
(defun msg fexpr (message)
  (progn
    (pr# 0)
    (print (eval (cons 'list message)))
    (pr# *armslot)))
```

To simplify the initialization process, the function `arminit` prompts for the interface card slot number and requests that the robot arm be positioned in some standard initial position.

```
(defun arminit ()
  (prog ()
    (print '(What slot is the robot card
              in?))
    (setq *armslot (read))
    (print '(enter the delay between motor
              steps))
    (setq *delay (read))
    (print '(using 123456/qwerty keys, set
              arm to initial position, 0 to exit))
    (pr# *armslot)
    (setarm)
    (resetarm) ;zero joint counts
    (pr# 0)
    (return 'arminit)))
```

MOVING THE ARM

At the present level of control it is possible to do several primitive things: move each joint of the arm some number of steps, read the status of the gripper switch, and read the recorded arm position. Another useful function at this time would be the ability to move the arm from wherever it is to another joint vector (remember, each joint vector uniquely defines an arm position). In terms of the step function, what is required is to step each joint some number of steps (positive or negative) until each joint reaches the desired position. In vector notation, if h is the current arm joint vector and p is the desired position, then the difference between the two vectors ($p-h$) should be applied to the step function. When this is done, the MiniMover-5 arm will not follow a straight line but will move through a shallow arc. A subfunction for performing vector subtraction is defined to simplify the vector calculations. The function `subvec` takes two lists of (assumed) equal length, a and b of (assumed) integers and returns the difference vector ($a-b$).

```
(defun subvec (a b)
  (cond ((null a) nil) ;end of vector
        (t (cons (sub (car a) (car b)) ;piece together
                  a-b
                  (subvec (cdr a) (cdr b)))) ;repeat on rest of list
```

The vector addition function is similar.

```
(defun addvec (a b)
  (cond ((null a) nil)
```

```
(t (cons (+
          (car a) (car b))
          (addvec (cdr a) (cdr b))))))
```

While those definitions look nice, the following iterative solution might run faster on some Lisp implementations.

```
(defun subvec (a b)
  (prog (ans)
    ck: (cond ((null a) (return ans))) ;end of vector
          (setq ans (append ans (list (sub
                                         (car a) (car b))))
          (setq a (cdr a)) ;remove first components
          (setq b (cdr b))
          (go ck:)))
```

Once the vector subtraction function is defined, the function to move the arm between two points is trivial.

```
(defun goarm (placevec)
  (prog ()
    (cond ((eq (steparm (subvec placevec
                               *here)) t) t) ;move ok?
          (t (error 'armgo: 'step-fails))) ;no to
    (setq *here placevec) ;update *here
    (return t)) ;confirm move
```

Function `goarm` is given a list `placevec` for its single input argument and attempts to move the arm to that position. If a failure is reported by the step function (this will never report failure on the MiniMover-5 since there is no joint position feedback), then the error routine is entered. The error routine prints its argument, evaluating unquoted elements, and breaks. The Lisp break function returns control to the user, but saves the evaluation stack. This means that when function `break` is called, you can evaluate what happened. When you evaluate the return command, evaluation continues after returning from the break function. If you are unable to correct the problem, the `reset` function can be used to clear the stack and return to the top level of the program. One possible error routine is defined as follows:

```
(defun error fexpr (x)
  (progn
    (msg (car x)) ;print error msg
    (break))) ;return control to user
```

NEXT MONTH

These basic arm commands can be used for low-level robot arm control. Next month, Part II of this article describes routines that allow you to reference positions and objects by names instead of position vectors.

MiniMover-5 is a registered trademark of Microbot, Inc. Apple II is a registered trademark of Apple Computer Corp.

Reader Feedback

To rate this article, circle the appropriate number on the Reader Service card.

70	80	90
Excellent	Good	Fair

INTRODUCTION TO PROLOG

Part I: The Commands

Dean A. Schlobohm

Micro-AI

PO Box 91

Rheem Valley, CA 94570

Prolog is gaining popularity throughout the world as an alternative to Lisp for symbolic computation and artificial intelligence programming. Since its development around 1970, Prolog has been used to write programs in such areas as natural language understanding, expert systems, and symbolic mathematics.

Recently, the Japanese announced that they are using Prolog as the core language for their Fifth Generation computers. With this announcement, people in the United States began considering Prolog as a viable alternative to Lisp for writing artificial intelligence programs.

Prolog stands for PROgramming in LOGic. As you will see from the following discussion, the language is a first step toward a goal of programming directly using logical constructions. Newcomers to Prolog will find the Prolog programs are entirely different from programs written in conventional languages such as BASIC, Pascal, and C. In particular, a Prolog programmer describes known facts about a given problem and the rules between these facts. Then, using a database of such facts and rules, the programmer can ask questions about the facts and rules. Moreover, Prolog can *infer* facts that are not expressly in the database from the rules and facts which are present.

Unlike programming languages such as BASIC, Pascal, and C, Prolog solves problems involving objects and relationships between objects. Generally, Prolog programming consists of the following steps:

- First, some *facts* about the problem in question are declared.

- Then, *rules* which govern the facts and relationships are defined.
- Finally, *questions* about the facts and their relationships are asked.

ADDING FACTS

Let's first discuss how to add facts about the problem in question to the computer's database. Suppose you want to add the fact that "John likes Pat" to your database. This fact consists of two objects, "John" and "Pat," and one relationship, "likes." In Prolog, the fact "John likes Pat" can be expressed as `likes(john,pat)`.

Note the following important things about the Prolog clause

```
likes(john,pat).
```

First, all the words start with a lowercase letter. In Prolog, names of specific objects start with lowercase letters while words that can stand for a number of specific objects (they are called variables) start with uppercase letters. Second, the relationship between the objects (i.e., *likes*) appears first. Throughout this article, this relationship is called the *predicate*. Third, the objects (i.e. `john` and `pat`) follow the predicate, are enclosed in parentheses, and are separated by a comma. Fourth, a clause which states a fact must be terminated by a period. Finally, the order in which the objects (i.e. `john` and `pat`) appear after the predicate is important. The example shown states, "John likes Pat." The statement "Pat likes John" would be represented by `likes(pat,john)`.

You can see that some consistent way of determining the order of the objects must be used. The *infix* representation is used throughout this article. To translate a Prolog clause into pseudo-English using infix notation means placing the predicate between the arguments. Thus, the clause `father(john,steve)` is translated into "John is the father of Steve," not "Steve is the father of John." Note that the predicate *father* here stands for the phrase "is the father of."

In Prolog, each name refers to a particular object. Thus, `john` refers to a single person, John. If your set of facts has two or more people named John, then you must distinguish them when you use them in your programming. For example, you could call them `john1` and `john2`. Furthermore, a given word can mean different things. For example, the word "bird" could mean a single bird as in the sentence "The bird is red" or it could mean the class of most birds as in the sentence "A bird can fly." Prolog does not distinguish the various definitions; a name has only one meaning. It is up to you to define your names so that each name has only one meaning. For example, you could express the fact that "The bird is red" by the Prolog clause `red(bird1)`.

The fact "A bird can fly" can be represented by `fly(bird)`. Here `bird1` represents a given bird while `bird` represents the class of most birds.

The names you use to represent objects and relationships are arbitrary as long as they start with a lowercase letter. Thus, the fact "John likes Pat" could be represented

as a(b,c). In this example, a represents likes, b represents John and c represents Pat. It is good programming practice, however, to use names which help you (and others trying to read your Prolog programs) remember what objects the names stand for.

ASKING QUESTIONS ABOUT FACTS

Once you have stored some facts in your database, you can ask questions about them. For example, assume your database contains the fact `likes(john,pat)`. You can ask the question "Does John like Pat?" by entering `likes(john,pat)?`

In response to this input, Prolog will answer `**yes` and then prompt you for more input. Prolog answers questions by searching through your database for a clause which *matches* the clause in question. The question matches a fact in your database if the predicate (i.e. `likes`) of the question matches the predicate of a fact in the database and each argument of the question matches the corresponding argument in the fact. Thus, in the above example, the question `likes(john,pat)?` matches the fact `likes(john,pat)`. since the predicate (`likes`) and the arguments (`john` and `pat`) are the same in each pattern.

If you ask the question "Does Pat like John?" (`likes(pat,john)?`), however, the computer answers `**no` since `pat` does not match `john`. What this means is that, as far as the computer knows, Pat does not like John. In other words, when Prolog responds with `**no` in answer to a question, it is telling you that the database does not contain a fact which matches the question. Thus, while it is possible that Pat does like John, the computer does not know this since you have not yet entered it into the database.

VARIABLES IN PROLOG

Up to now, all the facts and questions have contained specific objects. For example, you are not yet able to state, "Everybody likes John." To do this in Prolog requires the introduction of *variables*.

A variable is a special type of name which can match any object. In Prolog, variables must start with either an upper-case letter or the underline character. Thus, `A`, `John`, `STEVE`, and `_help` are all

variables. Variables are used for words like everybody, everyone, and everything. For example, "Everybody likes John" can be expressed as `likes(Everybody,john)`.

Variables can also be used in questions. For example, the question "Who likes John?" can be expressed as `likes(Who,john)?`

Questions in variables generally stand for words like who, what, and when. To see how this works, assume that the following facts about people who like John are in the database:

```
likes(albert,john).
likes(jeff,john).
likes(sally,john).
```

Now, if you ask the question, "Who likes John?" Prolog answers

```
Who = albert
Who = jeff
Who = sally
```

To understand how Prolog answers questions containing variables, we must first introduce some terminology. First, a variable is said to be *uninstantiated* when it does not stand for a specific object. For example, in the question `likes(Who,john)?`, the variable `Who` is uninstantiated before Prolog determines that Albert likes John. Once a variable stands for a given object, it is said to be *instantiated*. Thus, when Prolog sets `Who = albert`, `Who` is said to be instantiated to `albert`.

These definitions can be used to explain how Prolog answers questions containing variables. When Prolog is asked a question containing a variable, the variable is uninstantiated. Prolog then searches through the database for a clause that matches the question. In doing so, the variable matches any object. Thus, `likes(Who,john)` matches `likes(albert,john)` since the predicates are the same, `Who` matches anything and the second arguments are the same. When Prolog finds a match, it instantiates the variable to the name of the object which it matches. In the above example, `Who` became instantiated to `albert`.

Once a match is found, Prolog marks its place in the database. After printing the result, some versions of Prolog (e.g., Prolog-86) automatically return to this mark and seek another match. This process continues until all matches are found. This automatic attempt to find more

matches is called *backtracking* and is one of Prolog's important features.

QUESTIONS WITH AND & OR

Up to now, you have only learned how to ask simple questions about facts in your database. The next step is to ask questions containing `and` or `or`.

In Prolog, the comma (,) is an operator which stands for `and`. Thus, the question "Who likes John and Steve?" can be expressed as:

```
likes(Who,john) , likes(Who,steve)?
```

If your database contains the following facts

```
likes(jill,john).
likes(mary,john).
likes(mary,steve).
```

then Prolog will answer

```
Who = mary
```

In other words, Prolog finds that Mary likes both John and Steve. To reach this result, Prolog first attempts to satisfy (i.e. match) `likes(Who,john)` against the facts in the database. This match succeeds and `Who` is instantiated to `jill`. Prolog then attempts to match `likes(Who,steve)` where `Who` is instantiated to `jill` against the facts in the database. Since the fact `likes(jill,steve)` is not in the database, this match fails.

When the second match fails, Prolog automatically attempts to resatisfy the first clause `likes(Who,john)`. This time the match succeeds against the fact `likes(mary,john)`. with `Who` instantiated to `mary`. Prolog again attempts to satisfy `likes(Who,steve)` with `Who` instantiated to `mary`. This time the match succeeds since the fact `likes(mary,steve)` is in the database. Thus, Prolog answers by returning:

```
Who = mary
```

As you can see, the clause `A , B` is satisfied only if both `A` and `B` are satisfied.

Prolog also allows you to ask questions containing the word *or*. For example, to ask the question "Does Jill like John or Steve?" you would enter the clause `likes(jill,john) ; likes(jill,steve)?`

In Prolog, *or* is represented by either the semicolon (;) or the vertical bar (|). In answer to the above question, Prolog will answer `**yes` since Jill likes John.

However, what happens if you ask whether Jill likes Steve or Tim by entering the clause `likes(jill,steve) ; likes(jill,tim)?` Prolog answers ****no** since neither of the clauses are in the database.

When attempting to answer a question containing *or*, Prolog first attempts to match the leftmost clause in the question against the facts in the database. If the match succeeds, Prolog answers ****yes** or, if your question contains variables, Prolog returns the variable bindings. If the match does not succeed, Prolog attempts to satisfy the next clause. If this match succeeds, Prolog again answers ****yes** or returns the variable bindings. If the match does not succeed, the answer is ****no**. In other words, the clause `A ; B` succeeds if either A or B is satisfied.

ADDING RULES

One of the most powerful Prolog features is the ability to add rules, as well as facts, to your database. For example, assume that your database contains the following facts concerning the parents and sex of several individuals.

```
parent(dean_f,ann).
parent(dean_f,jan).
sex(ann,female).
sex(jan,female).
```

You can add the following rule to express whether a person is the sister of another person

```
sister_of(Person1,Person2) :-
    sex(Person1,female),
    parent(Parent,Person1),
    parent(Parent,Person2),
    diff(Person1,Person2).
```

```
diff(X,Y) :- X /= Y.
```

The first rule is interpreted as

*Person1 is the sister_of Person2 if
Person1 is sex female, and
some person Parent is the parent of
Person 1, and
Parent is also the parent of Person2,
and
Person1 is different from Person2.*

In Prolog, the operator `:-` stands for *if*. Also, the operator `/=` means *not equal to*.

After entering the above facts and rules, you can ask "Who is the sister of Ann?" by entering `sister_of(Who,ann)?` Prolog would answer `Who = jan`.

In other words, Prolog finds that Jan is the sister of Ann. To do this, Prolog first attempts to match `sister_of(Who,ann)` against the facts in the database. Since no fact of the form `sister_of(X,Y)` is in the database, this attempt fails.

Prolog next attempts to apply the rules in the database. Using the first rule, Prolog determines that Jan is the sister of Ann as follows. First, it attempts to match `sex(Who,female)` against the facts in the database. This match succeeds and instantiates `Who` to `ann`. Prolog then attempts to match `parent(Parent,Who)` with `Who` instantiated to `ann` against the facts in the database. Again this match succeeds and `Parent` is instantiated to `dean_f`. Next Prolog attempts to match `parent(Parent,ann)` with `Parent` instantiated to `dean_f` against the facts in the database. Again, this match succeeds. Finally, Prolog attempts to satisfy `diff(ann,ann)` using the second rule. Since `ann` is equal to `ann`, the second rule fails.

COMPUTER CONTROLLED ROBOTICS

1. DRIVER BOARD 5005 DB \$75 *
4.5" x 3.8" x 0.5", TTL/CMOS COMPATIBLE,
OPTICALLY INSULATED, FOR 4 PHASE MOTORS 2AMPS/50 VOLTS
 2. LINEAR ACTUATOR 601 AM \$75
12V/12W, 16 OZ, .001" STEP SIZE
19 LBS HOLDING FORCE, 3 IN TRAVEL
 3. LINEAR ACTUATOR 501 AM \$43
12V/3.5W, 1.5 OZ, .002" STEP SIZE
40 OZ HOLDING FORCE, 1.88 IN TRAVEL
 4. STEPPER MOTOR 201 SM \$16
5V/2W, 1.0 OZ, 15° STEP SIZE
0.8 OZ/IN HOLDING TORQUE
 5. STEPPER MOTOR 301 SM \$59
12V, 21.5 OZ, 1.8° STEP SIZE
80 OZ/IN HOLDING TORQUE
 6. MOTOR MOUNT FOR 301 SM \$25
 7. MOTOR MOUNT FOR 501 AM \$12
 8. MOTOR MOUNT FOR 501 AM \$13
- * EDGE CONNECTOR \$3.50



AMSI CORP. (516) 361-9499
BOX 651, SMITHTOWN, L.I., N.Y. 11787

TERMS Check, Money Order, C.O.D. VISA or MasterCard
Purchase Orders from Accredited Institutions.



After this failure, Prolog backtracks and attempts to resatisfy `sex(Person1, female)`. This time, the match succeeds with `Person1` instantiated to `jan`. As above, the next two clauses also succeed. Finally, `diff(jan,ann)` succeeds since `jan` is not equal to `ann`. As a result, the entire rule succeeds with `Who` instantiated to `jan`.

ARITHMETIC IN PROLOG-86

This section discusses the method used by Prolog to evaluate arithmetic expressions. Unlike most computer languages with which you may be familiar, arithmetic calculations take a back seat to pattern matching. At times, however, you will need to set a variable to the result of some arithmetic expression. For example, in Pascal the statement "`X := 3+4;`" sets `X` to seven. In Prolog, the `is` operator accomplishes the same result. The Prolog clause `X is 3+4?` causes `X` to be set to seven.

The `is` operator, however, is somewhat different than the assignment operator in languages such as Pascal and BASIC. For example, in Pascal the statement "`X :=`

`3+4;`" sets variable `X` to seven, even if `X` was previously set to another number. In Prolog, the clause `X is 3+4?` sets `X` to seven only if `X` is an uninstantiated variable. However, if `X` is an instantiated variable then the `is` operator first causes the expression on the right to be evaluated and then checks to see if the result matches the left side. For example, if `X` is instantiated to six, then `X is 3+4?` fails since seven (the result of `3+4`) does not match six.

To illustrate the use of arithmetic operators in Prolog rules, consider the rule for changing degrees Fahrenheit to degrees Celsius. If `F` is the degrees Fahrenheit and `C` is degrees Celsius, then `C = (F-32) (5 / 9)`. This rule can be expressed in Prolog as `f_to_c(C,F) :- C is (F-32) * 5 / 9.`

You can then ask the Celsius equivalent of 100 degrees Fahrenheit by entering `f_to_c(C,100)?` Prolog answers `C = 37.`

Note the round-off error that occurs since the division operator performs only integer division.

INTRODUCTION TO LISTS

Lists are a common data structure in sym-

bolic programming. A list is an ordered sequence of elements of any length. The fact that a list is an *ordered* sequence means that the order in which the elements appear is important. In Prolog, the elements of a list can be any atom, structure, or other term, including other lists. Lists in Prolog are very similar to lists in Lisp.

Generally, a list is either an atom representing the empty list (the list with no elements) or a structure with two arguments, a *head* and a *tail*. The head of a list is its first element. The tail of a list is the elements of the list with the head removed. In Prolog, the empty list is represented as `[]`.

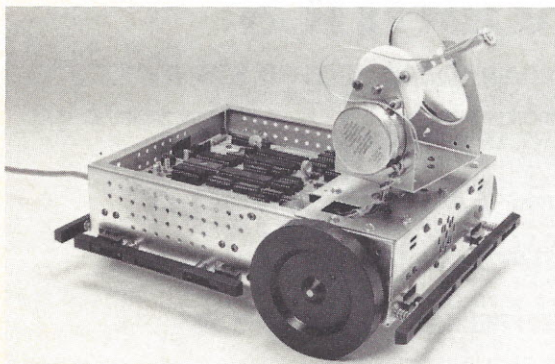
A list containing the three elements `dog`, `cat`, and `tiger` is represented as `[dog, cat, tiger]`. Since the order of the elements in a list is important, the list `[a, b, c]` is different from the list `[b, a, c]`.

Lists can also be used in structures. For example, assume your database contains the fact sentence(`[this,is,a,test,sentence]`). If you ask `sentence([H | T])?` Prolog responds

`H = this`

`T = [is,a,test,sentence]`

LIMITED TIME SPECIAL OFFER: SCORPION MOBILE ROBOT KIT



Regularly \$660
NOW ONLY \$299
SAVE OVER 50%

**FACTORY
DIRECT
ONLY**

The \$299 package price gives you the Scorpion kit with everything needed for assembly plus our 250-page assembly and operation manual (a \$20 value) as a free bonus.

The Scorpion is a versatile floor mobile robot that can be run from the RS-232C port of any computer, small or large. It is designed specifically for the serious robotic enthusiast who owns a personal computer and would like to experiment with robotics, artificial intelligence, machine intelligence, pattern recognition, and mobile robot theory. These activities are permitted by the Scorpion by virtue of its unique features, which include 2 drive motors, 4 bumpers with 2 microswitches each, 2 programmable eyes, a speaker with tone and duration programmability, a 2 axis optical scanner that allows patterns of the brightness of the robot's environment to be displayed on the CRT of the computer, and a 2 phase floor scanner that can be used to allow the Scorpion to track the floor environment in an intelligent manner.

World leaders in instructional robotics

RHINO
R O B O T S

3402 N. Mattis Ave.
P.O. Box 4010
Champaign, IL 61820
(217)352-8485

Enclose check or money order with order.
VISA and Mastercard charges accepted.


```

ask_about(has(X,Y,W,Z),T) :-
    print('Does the ',X,' have ',Y,'?'),
    !,nl,
    ratom(A),
    respond(A,has(X,Y,W,Z),T),!.
ask_about(gives(X,Y,W,Z),T) :-
    print('Does the ',X,' give ',Y,'?'),
    !,nl,
    ratom(A),
    respond(A,gives(X,Y,W,Z),T),!.
ask_about(eats(X,Y,W,Z),T) :-
    print('Does the ',X,' eat ',Y,'?'),
    !,nl,
    ratom(A),
    respond(A,eats(X,Y,W,Z),T),!.
ask_about(flies(X,Y,W,Z),T) :-
    print('Does the ',X,' fly ',Y,'?'),
    !,nl,
    ratom(A),
    respond(A,flies(X,Y,W,Z),T),!.
ask_about(swims(X,_W,Z),T) :-
    print('Does the ',X,' swim?'),
    !,nl,
    ratom(A),
    respond(A,swims(X,_W,Z),T),!.
ask_about(size(X,Y,W,Z),T) :-
    print('Is the ',X,' ',Y,'?'),
    !,nl,
    ratom(A),
    respond(A,size(X,Y,W,Z),T),!.
ask_about(breathing(X,Y,W,Z),T) :-
    print('Does the ',X,' breathe ',Y,'?'),
    !,nl,
    ratom(A),
    respond(A,breathing(X,Y,W,Z),T),!.
ask_about(vertebrate(X,_W,Z),T) :-
    print('Is the ',X,' a vertebrate?'),
    !,nl,
    ratom(A),
    respond(A,vertebrate(X,_W,Z),T),!.
    respond(why,X,[H|T]) :-
        respond(yes,X(A,B,[H|L],[D]),T) :-
            concat(['FACT: ',A,' ',X,' ',B],[D]),
            asserta(X(A,B,[D])).
        respond(why,X,[]) :-
            print("You should know!"),
            nl,
            ask_about(X,[]).
    respond(why,X,[H|T]) :-
        print(H),
        nl,!,
        ask_about(X,T).
has(A,B,L,M) :-
    ask_about(has(A,B,L,M),L),!.
gives(A,B,L,M) :-
    ask_about(gives(A,B,L,M),L),!.
eats(A,B,L,M) :-
    ask_about(eats(A,B,L,M),L),!.
flies(A,B,L,M) :-
    ask_about(flies(A,B,L,M),L),!.
swims(A,_L,M) :-
    ask_about(swims(A,_L,M),L),!.
size(A,B,L,M) :-
    ask_about(size(A,B,L,M),L),!.
breathing(A,B,L,M) :-
    ask_about(breathing(A,B,L,M),L),!.
vertebrate(A,B,L,M) :-
    ask_about(vertebrate(A,B,L,M),L),!.
isa(A,mammal,M,L) :-
    concat(['RULE 1: ',A,' is a mammal if it has
hair.'],Z),
    append(M,[Z],X),
    has(A,hair,X,W),!,
    append([Z],W,L),
    asserta(isa(A,mammal,M,L)).
isa(A,mammal,M,L) :-
    concat(['RULE 2: ',A,' is a mammal if it gives
milk.'],Z),
    append(M,[Z],X),
    gives(A,milk,X,W),!,
    append([Z],W,L),
    asserta(isa(A,mammal,M,L)).
isa(A,bird,M,L) :-
    concat(['RULE 3: ',A,' is a bird if it has
feathers.'],Z),
    append(M,[Z],X),
    has(A,feathers,X,W),!,
    append([Z],W,L),
    asserta(isa(A,bird,M,L)).
isa(A,bird,M,L) :-
    concat(['RULE 4: ',A,' is a bird if it
flies.'],Z),
    append(M,[Z],X),
    flies(A,well,X,W),!,
    append([Z],W,L),
    asserta(isa(A,bird,M,L)).
isa(A,carnivore,M,L) :-
    concat(['RULE 5: ',A,' is a carnivore if it eats
meat.'],Z),
    append(M,[Z],X),
    eats(A,meat,X,W),!,
    append([Z],W,L),
    asserta(isa(A,carnivore,M,L)).
isa(A,reptile,M,L) :-
    concat(['RULE 6: ',A,' is a reptile if is an
air-breathing vertebrate.'],Z),
    append(M,[Z],X),
    breathing(A,air,X,W),
    vertebrate(A,_X,N),
    append([Z],W,R),
    append(R,N,L),
    asserta(isa(A,reptile,M,L)).
isa(A,tiger) :-
    concat(['RULE 7: ',A,' is a tiger if it is a
mammal, ',\n', 'and a carnivore, and has black',
stripes.'],Z),
    isa(A,mammal,[Z],M),
    isa(A,carnivore,[Z],N),
    has(A,'black stripes',[Z],E),
    print(A,' is a tiger. '),nl,!,
    append(M,N,W),
    append(W,E,G),
    asserta(isa(A,tiger,[Z],[Z|G])).
isa(A,ostrich) :-
    concat(['RULE 8: ',A,' is an ostrich if it is a
bird, ',\n', 'has a long neck and long legs.'],Z),
    isa(A,bird,[Z],M),
    has(A,'a long neck',[Z],C),
    has(A,'long legs',[Z],E),
    print(A,' is an ostrich. '),
    nl,!,
    append(M,C,F),
    append(F,E,G),
    asserta(isa(A,ostrich,[Z],[Z|G])).
isa(A,whale) :-
    concat(['RULE 9: ',A,' is a whale if it is a
mammal, ',\n', 'has a gray color, swims and is huge.'],Z),
    isa(A,mammal,[Z],M),
    has(A,'a gray color',[Z],N),
    swims(A,_[Z],C),
    size(A,huge,[Z],D),
    print(A,' is a whale. '),
    nl,!,
    append(M,N,W),
    append(C,D,F),
    append(W,F,G),
    asserta(isa(A,whale,[Z],[Z|G])).

```

Continued

Listing 1. Continued

```
isa(A,robin) :-
    concat(["RULE 10: ",A," is a robin if it is a bird
    and has a red breast.",Z),
    isa(A,bird,[Z],M),
    has(A,"a red breast",[Z],N),
    print(A," is a robin."),
    nl,!,
    append(M,N,G),
    asserta(isa(A,robin,[Z],[Z[G]])).

isa(A,rattlesnake) :-
    concat(["RULE 11: ",A," is a rattlesnake if it is a
    reptile and has rattles.",Z),
    isa(A,reptile,[Z],M),
    has(A,"rattles",[Z],N),
    print(A," is a rattlesnake."),
    nl,!,
    append(M,N,G),
    asserta(isa(A,rattlesnake,[Z],[Z[G]])).

how(X(A,B)) :-
    X(A,B,_,[H|T]),!,
    print(H),
    !,
    howl(T),!.
howl([]) :-
    print("END OF EXPLANATION."),
    !.
howl([H|T]) :-
    print(H),
    howl(T),!.
diagnose :-
    print("This program will help you diagnose which of
    the "),
    print("following animals you are thinking of:"),
    nl,
    print("tiger, ostrich, robin, whale, or
    rattlesnake"),
    nl,nl,
    possibilities(X),!,
    diagnose(X),!.
diagnose([]) :-
    print("No hypothesis can be confirmed."),
    !.
diagnose([H|T]) :-
    isa(Animal,H),!,
    nl,
    ask("Do you want to know how I reached this
    conclusion?",A),!,
    A = yes,
    nl,
    how(isa(Animal,H)),!.
diagnose([H|T]) :-
    !,
    diagnose(T).
possibilities([robin, rattlesnake, ostrich, whale,
    tiger]).
append([],L,L).
append([Z|L1],L2,[Z|L3]) :-
    append(L1,L2,L3).
```

list's tail, Y. Note that this rule defines member in terms of itself. This is called a *recursive* procedure definition.

After you have defined member, you can determine if an element is a member of the set by asking member(a, [a,b,c])? In this case, Prolog answers ****yes** using the first rule.

Similarly, if you ask member(X, [a,b,c])? Prolog answers

```
X = a
X = b
X = c
```

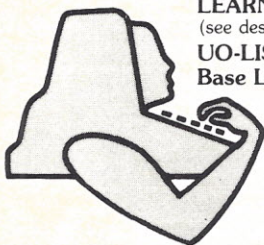
In this example, Prolog successively set X to each member of the list [a,b,c].

MODIFYING THE DATABASE

Prolog provides several built-in procedures for adding or removing clauses from your database. For example, if your database contains the rule that all mammals have hair, and you later learn that tiger1 has hair, you may want to add the fact tiger1 is a mammal to your database. This may speed up future searches.

(LISP) FOR A.I.

UO-LISP Programming Environment The Powerful Implementation of LISP for MICRO COMPUTERS



LEARN LISP System (LLS.1)
(see description below) **\$39.95**
UO-LISP Programming Environment
Base Line System (BLS.1) **\$49.95**

Includes: Interpreter, Compiler, Structure Editor, Extended Numbers, Trace, Pretty Print, various Utilities, and Manual with Usage Examples. (BLS.1) expands to support full system and products described below.

UO-LISP Programming Environment: The Usual LISP Interpreter Functions, Data Types and Extensions, Structure & Screen Editors, Compiler, Optimizer, LISP & Assembly Code Intermixing, Compiled Code Library Loader, I/O Support, Macros, Debug Tools, Sort & Merge, On-Line Help, Other Utility Packages, Hardware and Operating System Access, Session Freeze and Restart, Manual with Examples expands to over 350 pages. Other UO-LISP products include: LISPTEXT text formatter, LITTLE META translator writing system, RLISP high level language, NLARGE algebra system. Prices vary with configurations beyond (BLS.1) please send for **FREE** catalog.

LEARN LISP System (LLS.1): Complete with LISP Tutorial Guide, Editor Tutorial Guide, System Manual with Examples, Full LISP Interpreter, On-Line Help and other Utilities. LEARN LISP fundamentals and programming techniques rapidly and effectively. This system does not permit expansion to include the compiler and other products listed above.

LISP Tutorial Support (LTS.1): Includes LISP and Structure Editor Tutorial Guides, On-line Help, and History Loop. This option adds a valuable learning tool to the UO-LISP Programming Environment (BLS.1). Order (LTS.1) for **\$19.95**.

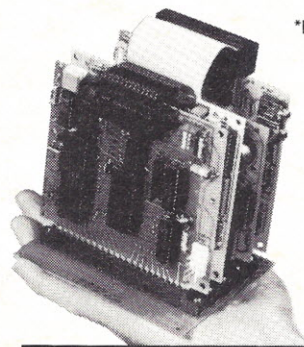
REQUIRES: UO-LISP Products run on most Z80 computers with CP/M, TRSDOS or TRSDOS compatible operating systems. The 8086 version available soon.

TO ORDER: Send Name, Address, Phone No., Computer Type, Disk Format Type, Package Price, 6.5% Tax (CA residents only), Ship & Handle fee of \$3.00 inside U.S. & CN, \$10 outside U.S., Check, Money Order, VISA and MasterCard accepted. With Credit Card include exp. date. Other configurations and products are ordered thru our **FREE** catalog.

Northwest Computer Algorithms
P.O. Box 90995, Long Beach, CA 90809 (213) 426-1893

Micromint will put both a computer development system and an OEM dedicated controller in the palm of your hand for as little as \$105*

*In quantities of 100



Z8 BASIC SYSTEM CONTROLLER
(BCC11) .. \$149

NEW Z8 FORTH SYSTEM CONTROLLER .. (BCC21) .. \$280

NEW TERM-MITE Z8 BUS COMPATIBLE INTELLIGENT TERMINAL BOARD .. (BCC22) .. \$284

Z8 Expansion Board
w/8K memory .. (BCC04) .. \$180

Z8 Mother Board

w/ 5 gold connectors .. (MB02) .. \$ 69

Z8 EPROM Programmer .. (BCC07) .. \$145

Z8 Serial Expansion Board .. (BCC08) .. \$160

Z8 BASIC A to D Converter (BCC13) .. \$140

Z8 BASIC 16K Memory Expansion
(BCC15) .. \$155

Z8 CROSS ASSEMBLERS

FROM ALLEN ASHLEY
For TRS-80 Model I (XAS01) .. \$ 75
For TRS-80 Model III (XAS02) .. \$ 75
CP/M 2.2 8" (XAS03) .. \$150
Northstar 5 1/4" (XAS04) .. \$ 75

FROM MICRO RESOURCES
CP/M 2.2 8" (MR01) .. \$ 75
Apple II CP/M 5 1/4" (MR02) .. \$ 75
6502 based systems (MR03) .. \$ 75
IBM PC 5 1/4" (MR04) .. \$ 75



To Order:

Call Toll Free

1-800-645-3479

For Information Call:

1-516-374-6793



MICROMINT INC.
561 Willow Avenue
Cedarhurst, NY 11516

Z8 is a trademark of Zilog Inc.
CP/M is a trademark of
Digital Research

Advertising

Reader Service

R.S.

No. Company

- 1 Amrobot, p. 29
- 2 AMSI Corp., p. 15
- 3 Artec Systems, Cover II
- 4 Automation News, p. 21
- 5 Centroid, p. 32
- 6 Control Ware, p. 32
- 7 Creative Learning Systems, Inc., p. 3
- 8 Cyber Robotics, p. 29
- 9 Cybot, Inc., p. 32
- 10 Cybot, Inc., Cover IV
- 11 Denning Mobile Robotics, p. 35
- 12 Digi-Tech, Inc., p. 33
- 13 Dr. Dobb's Journal, p. 5
- 14 Grant Publications, Inc., p. 23
- 15 HoB-Bots of Hudson, p. 30
- 16 James E. Iannoni & Assoc., p. 30
- 18 Laboratory Microsystems, Inc., p. 6
- 19 MicroMint, Inc., p. 18
- 20 MicroMotion, p. 2
- 21 Micron Technology, Inc., p. 9
- 22 Northwest Computer Algorithms, p. 18
- 23 Oriental Motor USA Corp., p. 9
- 24 Personal Robotics Corp., p. 4
- 25 Personal Robots Magazine, p. 30
- 26 Prep, Inc., p. 11
- 17 The Report Store, p. 31
- 27 Rhino Robots, Inc., p. 16
- 28 Rio Grande Robotics, p. 33
- 29 Robotic Microsystems, p. 31
- 30 Sabadia Export Corp., p. 7
- 31 Schulz Enterprises, Inc., p. 31
- 32 Schvien and Son, p. 33
- * Sensors, Cover III
- 34 Software Science, p. 33
- 35 Space Robot Market, p. 32
- 33 Spectron Instrument, p. 25
- 36 Tech Books, p. 30
- 37 Vernitech, p. 31

*Correspond directly with company

In Prolog, the procedures `asserta` and `assertz` add new clauses to your database. The predicate `asserta` adds the new clause at the beginning of your database and `assertz` adds the clause at the end. To add the fact that `tiger1` is a mammal at the beginning of your database, your program could execute the procedure `asserta(mammal(tiger1))`!

As a further example, let's define a procedure `add` that adds a clause to your database only if the clause is not already there. This is useful in preventing your database from becoming filled with repetitious data. `add` is defined as:

```
add(X) :- X,!.
add(X) :- asserta(X).
```

The first rule of `add` states that if `X` exists in the database, do not add it. If `X` does not exist in the database (i.e. `X` does not match any clause in your database), then the second rule of `add` uses `asserta` to add the fact at the beginning of your database.

The Prolog procedures `retract` and `retractall` are used to remove clauses from your database. For example, if you want to remove all clauses of the form `mammal(X)` from your database, you would enter the command `retractall(mammal(X))`!

Lists are useful for representing English sentences since each word in the sentence can be easily accessed.

Now, let's take all the information learned so far and write some simple Prolog procedures. Suppose you want to find out if an atom or other structure is an element of a list. It would be convenient to have a procedure that answers `**yes` if the atom or structure is an element of the list and `**no` otherwise. Let's define a pro-

cedure called `member` such that `member(A,L)?` returns `**yes` if `A` is an element (a member) of list `L`. To define the procedure `member`, you should recognize two rules:

- `A` is a member of list `L` if `A` is the first element of `L`.
- If `A` is not the first element of list `L`, then `A` is a member of `L` only if `A` is a member of the tail of the list `L`.

The first rule is easily expressed in Prolog as `member(A,[_|_])`. This clause is satisfied only if `A` is the first element of the list `[_|_]`. (The underline character is a variable that matches anything.)

You can express the second rule in Prolog as `member(A,[_|Y):-member(A,Y)`. This Prolog rule means that `A` is a member of the list `[_|Y]` if it is a member of the

NEXT MONTH

Although these are just the very basic commands and concepts for programming in Prolog, they provide you with sufficient power to experiment with expert system development. Next month, Part II shows the basic concepts for creating a simple expert system.

The version of Prolog discussed in this article is Prolog-86, an implementation of Prolog currently available for microcomputers running the MS-DOS and CP/M-86 operating systems. Prolog-86™ is a trademark of Micro-AI. MS-DOS™ is a trademark of Microsoft. CP/M-86™ is a trademark of Digital Research.

Reader Feedback

To rate this article, circle the appropriate number on the Reader Service card.

71	81	91
Excellent	Good	Fair

Classified Advertising

ATTORNEY-BUSINESSMAN SEEKS ENGINEERS—for mobile robot venture with related expertise in sensors, power drives, computer hardware/software and short-range navigation. Call (212) 997-1040.

COMPUTER MOTION CONTROL—learn the principles by building a flatbed plotter controlled by a Commodore 64 or Vic 20 computer. Plans, programs, manual, etc., \$49.00. Kit including motors \$169.00, fully assembled \$249.00. Stepper motor controller which controls two motors for the Com-

modore 64 or Vic 20 with fast machine code program on disk or tape \$75.00, with the motors \$99.00. Maxiplot, 839 Seafoam, Houston, TX 77062.

PERSONAL ROBOTS, KITS and BOOKS available at Cal Robots 16200 Ventura Blvd., Ste. 223, Encino, CA 91436, (818) 905-0721.

FOUR FINGER HAND & WRIST: Three moving fingers, one stationary, opens 3" 90° wrist-flexion, needs only two actuator inputs! Write San Fran Hands, PO Box 881972, San Francisco, CA 94188-1972.

EARL

An Easy Robot Language

D. S. Holmes
Stochos, Inc.
14 North College St.
Schenectady, NY 12305

There is an increasing use of light-duty, low-cost robots in industry. Several examples of such robots are the Teachmover™ (Microbot), the XR-II™ (Rhino Robots), and the Excalibur™ (Robotic Systems International). This class of robots allows automation to be applied to tasks that do not justify the investment in larger, more expensive units.

These light-duty robots can, as their larger cousins, be taught by moving them through the desired sequence of motions and recording their positions and appropriate commands in the onboard computer. However, there are several limitations inherent in this approach:

- The robot must be nonproductive while it is being taught.
- The onboard computers have rather limited memory capacity which, in turn, limits the complexity of the tasks that can be performed.
- Coordinating the moves of several robots is difficult.

The ability to talk to these robots via the ubiquitous personal computer helps to overcome these limitations. The language used for communication should:

- be simple to use
- be easy to remember
- work on a wide variety of personal computers
- have a cost proportional to the robot and personal computer investment

One possible solution is to furnish an extended version of BASIC—such as Microbot's Armbasic™. The main drawback of this approach is that the operator must learn about computer programming before trying to instruct the robot. Thus, using

BASIC, simple as it is, diverts the user from his task.

Another solution is to furnish a *high-level* robot language that eliminates the necessity for learning BASIC. This language should be designed specifically for defining robot tasks. For example, if you desire to have the robot go to the XYZ location (5,6,7.3), then the language should accept a command similar to "GOTO X=5 Y=6 Z=7.3" or "MOVETO X5,Y6,Z7.3". If the action is to be conditional upon the width of the item in the gripper, then the language should accept

something like "IF GRIP >25 THEN GOTO X=5 Y=6 Z=7.3".

EARL (EAsy Robot Language) is an example of such a language. This language was developed by Peter Holmes of Stochos, Inc. for use with the Microbot Teachmover. To write programs in this language, you simply use a standard text editor or word processor to enter a list of commands into a text file. The EARL compiler then operates on the text file, runs syntax checks and, if no errors are detected, produces a file containing the alphanumeric codes required to control the robot. If er-

{TEXT}	EARL INPUT	DATA TRANSMITTED TO RUN THE MICROBOT
ACTION DESIRED		
Initialize the Microbot	INIT	@DELAY 5 @SET 200 @RESET @CLOSE 200 @READ, then decipher returned data
Set motor speed to 235	SPEED 235	@SET 235
Move to XYZ coordinate (5,0,3)	MOVE 5 0 3	@STEP 235 0 -58 -262 1 1 -262
Open the gripper	GRIP OPEN	@STEP 235 0 0 0 0 0 742
Move to XYZ coordinate (8,3,3)	MOVE 8 3 3	@STEP 235 -404 250 -236 -88 88 -236
Move to XYZ coordinate (8,3,0)	MOVE 8 3 0	@STEP 235 0 446 167 0 0 167
Close the gripper tightly and determine width of held object	GRIP TIGHT	@CLOSE 235 @STEP 235 0 0 0 0 0 -32 @READ, then decipher returned data

Listing 1. A comparison between EARL commands and the commands required by a Microbot robot system. The EARL commands are much simpler to use and understand.

STATEMENT SYNTAX	USE
!	Comments. Anything may follow the exclamation point. A comment may be either on a line by itself or appended to the end of a line.
INIT	Initialize the robot. Must be the first executable line in your program.
PITCH angle	Set the pitch angle of the robot's wrist. Valid angles are -0.1 degrees thru -179.9 degrees.
ROLL angle	Set the roll angle of the robot's wrist. Valid angles are -90 degrees thru 90 degrees.
SPEED speed	Set the speed at which the robot's motors travel. Valid speeds are 0 thru 245. Note that the robot is generally inaccurate at speeds greater than 238.
MOVE xloc yloc zloc	Move the robot to an XYZ location.
GRIP TIGHT CLOSE OPEN number	Set the location of the robot's gripper. GRIP TIGHT allows the robot to pick up 1 pound. GRIP CLOSE just closes the gripper. GRIP OPEN opens the gripper to 2 in. GRIP number opens the gripper to "number" inches.
IF GRIP < number THEN line <= = >= > <>	Provides for conditional branching depending on the gripper position. (Due to potential accuracy problems, it is suggested that equality (= and <>) are probably not appropriate for use with this class of robots.)
GOTO line	Provides for unconditional branching.
DO number TIMES NOMORE	Provides for looping. DO number TIMES will do everything up to DO NOMORE as many times as specified. DO loops may be nested a maximum of eight deep.

Table 1. The complete set of EARL commands and functions.

rors are detected, error messages are sent to the user. Once a robot-readable file is built on the computer, the instructions are passed to the robot and it performs the desired actions.

Listing 1 shows an example of the natural advantage provided by higher-level languages like EARL. Notice the vast difference between the simple EARL commands and the information transmitted to the Microbot robot.

A syntax summary for EARL is shown in Table 1. Items enclosed in vertical bars indicate that a choice must be made. Items in uppercase letters must be typed exactly as shown. Items in lowercase letters represent a numeric value (decimal format, e.g. 1.2E5 is *not* acceptable).

Notice in both Listing 1 and Table 1 the power of EARL to:

- branch to a large variety of robot tasks

depending on the condition of the gripper

- repeat a set of tasks for a fixed number of cycles

Thus, the language and the personal computer together add greatly to the capability of the robot. As personal computers move forward into the multitasking world, the activities of whole sets of these robots can be coordinated. Once you start using a high-level robot language, you will find a dramatic improvement in your productivity and the productivity of your robot.

Reader Feedback

To rate this article, circle the appropriate number on the Reader Service card.

72	82	92
Excellent	Good	Fair

AUTOMATION NEWS

The Newspaper for Flexible Manufacturing June 4, 1984 A Great Business Publication Vol. 2, No. 6

Republicans Release Program To Spur High Tech Innovation

By Kenneth Jones
Washington Bureau

WASHINGTON, June 3—The House of Representatives today released a program to spur high tech innovation. The program, which is part of a larger effort to encourage research and development in the private sector, will provide tax incentives for companies that invest in research and development. The program is expected to be passed by the House in the near future.

House Bill Eases Joint R&D Risk

By Kenneth Jones
Washington Bureau

WASHINGTON, June 3—The House of Representatives today passed a bill that would ease the risk of joint research and development (R&D) between a company and a government agency. The bill, which is part of a larger effort to encourage research and development in the private sector, would allow a company to claim a tax credit for its share of the R&D costs. The bill is expected to be passed by the House in the near future.

Panasonic Enters Pana Robo In U.S.

By Kenneth Jones
Washington Bureau

WASHINGTON, June 3—Panasonic Corp. today announced that it had entered the U.S. market with its new robot, the Pana Robo. The robot is a compact, programmable robot that is designed for use in a variety of manufacturing applications. The robot is expected to be available in the U.S. market in the near future.

Toyoda Regains Machining Ctr's

By Kenneth Jones
Washington Bureau

WASHINGTON, June 3—Toyoda Machine Works today announced that it had regained its status as a leading manufacturer of machine tools. The company, which had previously been forced to leave the U.S. market, has now been allowed to return. The company is expected to be a major player in the U.S. market in the near future.

IR/1 P256 Building Block For GMF Vision System

By Kenneth Jones
Washington Bureau

WASHINGTON, June 3—The IR/1 P256 Building Block, a new addition to the GMF Vision System, was announced today. The building block is a compact, programmable unit that is designed for use in a variety of manufacturing applications. The building block is expected to be available in the U.S. market in the near future.

SEE THE NEWEST 3 AXIS ROBOT AT ROBOT VILL.

By Kenneth Jones
Washington Bureau

WASHINGTON, June 3—The newest 3 axis robot, the Milacron 3000, was announced today. The robot is a compact, programmable robot that is designed for use in a variety of manufacturing applications. The robot is expected to be available in the U.S. market in the near future.

New Subscriber Offer Features Reduced Rate Along With Free Gift

NEW YORK—Manufacturing management and engineers can now receive AUTOMATION NEWS at a reduced, new subscriber, rate of \$18.75 along with an 80 page, factory management glossary as a gift.

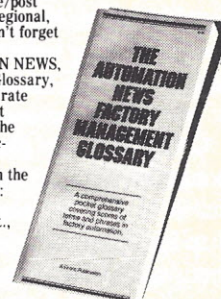
This introductory offer will save you \$9.25 off the regular subscription rate of \$28. In addition, if you become a subscriber for a full year or more we'll ship you, immediately on payment, a comprehensive, cross-referenced, perfect bound, factory management glossary containing over 2,500 terms in advanced manufacturing technologies covering categories from CAD/CAM to Vision, robots to CIM, and components to programmable controllers.

Every issue of AUTOMATION NEWS will inform you through objective easy-to-read reporting, of the latest factory automation developments in CAD/CAM, computers, robots, FMS, components, lasers, programmable controllers, software, vision and sensors, reports from abroad, association news, finance (complete with stock listings and insider trading), people on the move, industry surveys, pre/post show coverage on a local, regional, and national basis. And, don't forget our classified section.

Combined: AUTOMATION NEWS, the Factory Management Glossary, and a reduced subscription rate make this an offer you can't refuse if you want to be in the know on advanced manufacturing technologies.

To subscribe simply fill in the coupon below and mail it to:
AUTOMATION NEWS,
Subscription Service Dept.,
P.O. Box 1141,
Dover, NJ, 07801
or call Mike Ryan
for faster service
at 212-223-0232.

To get your glossary immediately, pull out your credit card. We accept Master Charge and VISA.



Yes please enter a subscription in my name and bill me at the new subscriber rate of \$18.75 (33% off the regular yearly rate of \$28)

I can't wait! Enclosed please find my check for \$18.75. Please enter my subscription and rush my factory management glossary.

Name _____ Title _____

Company _____

Address _____

City _____ State _____ Zip _____

Type of Business _____

If mfg., product produced? _____

of employees at location _____

Signature _____ Date _____

Please allow 4-6 weeks for service to start. All questions must be answered to qualify. If at any time you are not satisfied you can cancel and receive a refund for all issues not mailed; the glossary is yours to keep.

3RA345

THE STRING COMPUTER

Russ Adams
3008 Mosby St.
Alexandria, VA 22305

Computers are exact, and there's the rub to artificial intelligence. Intelligence requires the ability to find similarity rather than equivalence. When a computer is asked to supply information based on an inexact, incompletely, or inaccurately phrased question, it usually supplies the wrong information. However, people perform many common tasks correctly with only sketchy information. Common tasks like finding a name in a list, checking for correct spelling, or understanding a natural language sentence draw upon a person's ability to match similar patterns.

Computers have been made to mimic this fuzzy matching capability with help from special algorithms. Since these algorithms have been implemented in software, they have been slow and limited in ability, and have taken up large amounts of computing resources.

Generally, these algorithms have functioned by first limiting their search for a match to a small number of strings and then identifying the most similar strings of that small number. Limiting the search to a small number of strings was required because string-matching algorithms have only been able to make a few thousand comparisons a second. However, limiting the search will limit its effectiveness; though it may work fine with string differences up to a certain level, beyond that level the search algorithm may fail.

Spelling-checking algorithms, for example, have required words that are misspelled by no more than three deletions, insertions, or substitutions of letters. Take as an example a comparison of *shartroose* and *chartreuse*. A spelling-checking algorithm with the above limits would fail to recognize the similarity between the strings. Yet the similarity of the two words

is apparent. The ability to easily notice such similarity is a unique characteristic of human intelligence. Such a characteristic has been difficult to duplicate using a computer—until now.

AN INTELLIGENCE CHIP

Early in 1984, Proximity Technology, Inc. of Fort Lauderdale, Florida introduced a VLSI chip designed to bring computer technology one step closer to duplicating human intelligence. The PF474 is a VLSI integrated circuit that can perform a powerful string comparison function at speeds as high as 40,000 comparisons per second for 8-character strings. At such speeds, many search problems can be solved by brute force in acceptable times.

The string comparison function used in the PF474 is very flexible and can be adapted to many different searching problems. Such flexibility is important. For example, checking the spelling of English words requires a different comparison method than comparing proper names or French words. The PF474 takes a search or query string and compares it with all other strings in a database. For each comparison, a *proximity value* is computed which is related to how closely the strings match. The highest values (corresponding to the best matches) are stored in a ranked list for reference at the end of the search.

The proximity value, after a word *A* is compared with a word *B*, can be expressed as:

$$P.V. = \frac{2 \times (A \text{ compared to } B)}{(A \text{ compared to } A) + (B \text{ compared to } B)}$$

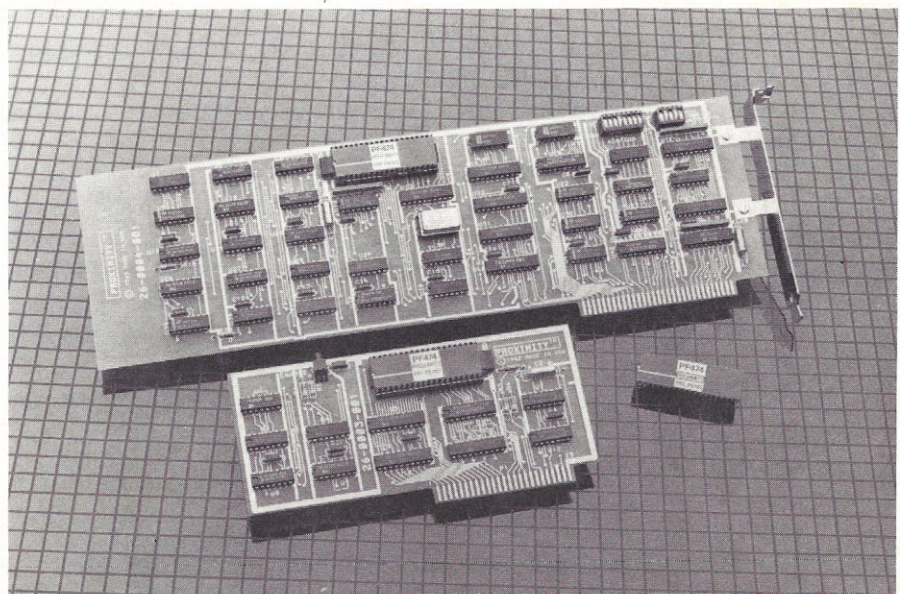


Photo 1. The PF474 proximity chip from Proximity Technology, Inc. can be used to search for near matches in two strings. At present, Proximity Technology has also designed pattern-matching boards for the IBM PC and Apple II computers.

MANAGEMENT BOOKS-BY-MAIL NATIONAL BESTSELLERS

A Special Grant Publications Book Service For Manufacturing Managers

Circle 14

BASIC Programming Solutions for Manufacturing

By J. Nicks

Explore 22 tested programs that will produce easily understood reports on the topics faced by Manufacturing Engineers everyday. Explore chapters on learning curves and launching costs, machine capability studies, equipment justification, manpower analysis, cost estimating, tolerance control, and many more.

In addition, these programs were written and tested on a personal "desk-top" microcomputer.

Finally, enough information and explanation is included on each program so that you can tailor them to your own needs or write totally new programs. 250 pp. 1981.

Hardcover: \$32.00

An Introduction to Robot Technology

By Philippe Coiffet and Michael Chirouse

Aimed at the engineering and computer professional, this book combines electronics and mechanical engineering topics to provide a practical introduction to the basic principles and concepts of robotics. Numerous mathematical models, diagrams, and appendices illustrate typical robot design considerations along with individual components such as controls, motors, actuators, and sensors. Emphasizing applied concepts as well as theoretical, the book also includes chapters on the computer control, characteristic properties, usage, "training," and real-world applications of robotics.

6 1/2 x 9 1/4 208 pp. 60 illus.

\$29.95

Robotics Research: The First International Symposium

Edited by Michael Brady and Richard Paul

The fifty-three contributions collected in this book present leading current research in one of the fastest moving fields of artificial intelligence. Organized around a view of robotics as "the intelligent connection of perception to action," they convey the excitement of cross-disciplinary discussion by scholars from the United States, Japan, France, the United Kingdom, West Germany, and Australia.

Chapters in the book's first part explore the connection between perception and action in three sections that deal with task level programming, integrated systems, and walking machines. The second part reports recent progress on the perceptual basis of robotics, with chapters grouped in sections on visual inspection, three-dimensional vision, and (nonvisual) local sensing. The third part focuses on systems that facilitate action, with sections that discuss mechanisms, kinematics and dynamics, and feedback control. A final part considers the application of robot systems to manufacturing, with chapters divided into two sections: on systems for manufacture and on robots and manufacture. The editors have written introductions to each of the book's four major parts and eleven sections.

This book is the twelfth in the MIT Press Series in Artificial Intelligence, edited by Patrick Henry Winston and Michael Brady.

7 x 10, 1000 pp., 600 illus.

\$65.00

New Edition!

The practical guide to numerical control...

Principles of Numerical Control

3rd Edition

By James J. Childs

The most comprehensive, clearly written introduction to the operation of numerical control systems available for the non-expert written by the president of a numerical control consulting organization. This widely used, up-to-date reference provides all the information necessary for anyone—managers, students, supervisors, engineers—to gain a working understanding of its functions.

The emphasis is on CNC Systems and their programming capabilities, advantages, operation, and maintenance. Yet, a good portion of the hardware material from past editions has been retained to satisfy those systems still in operation. A wide range of topics covers everything from specific information on plant requirements, to step-by-step economic justification procedures (with tables listing comparative savings versus conventional hardware equipment), to detailed part programming and basic computer operations. The latest in EIA Standards describe updated and standardized tape codes for easy reference.

Reflecting advances and future trends in numerical control, the book presents meticulously updated discussions of such topics as:

- Robotics
- Graphics
- Group Technology
- DNC

Numerous examples of both APT and COMPTACT II part programming provide a firm basis for more advanced programming courses for students.

1982, 300 pp., illus., \$24.95

Your one-stop guide to the computer graphics marketplace

The 1984 S. Klein Directory of COMPUTER GRAPHICS SUPPLIERS

Hardware, Software, Systems and Services

This latest edition identifies more than 500 supply sources. Directory entries provide basic product information and business background on each company, including ownership, top management, company size, sales volume, and year of origin as well as address, phone number, telex or TWX, and persons to contact. The Directory also features a computer generated, cross-index that enables readers to identify vendors by applications of interest and by specific technology. In addition, you also get a 5-page briefing, prepared by Carl Machover, industry consultant, on the considerations and tradeoffs to be made when buying computer graphics products. 224 pp.

\$50.00

Industrial Robotics Handbook

By V. Daniel Hunt

In one comprehensive volume, this extensively illustrated handbook provides a unique overview of the subject of robotics, its hardware, various types, their functions and future in industry. All of the information presented has been obtained from a wide variety of knowledgeable sources in the field including the most recent research available. An extensive selection of photographs, diagrams, and charts amplify the text.

Hunt reveals a wealth of information on foreign research and development in the industry and the companies involved in and exploring the manufacture of industrial robots. He also supplies an extensive list of the various robotic systems, including the potential of "smart robots," grouped into types of models. The list includes important technical information on specific applications pertaining to: tolerances, load carrying capacities, price, and names and addresses of companies and individuals to contact for further information.

1983, 352 pp. (T), illus. \$32.50 (T)

Industrial Robotics Handbook provides specific information on:

- Sensor Systems
- Manipulator Hardware
- Dynamic Properties
- Technology Status
- Tooling
- Applications
- Control Systems
- Programming
- Safety Considerations
- Socioeconomic Impacts of Robotics
- ...and much, much more

The Professional Guide to Automatic Assembly... From the Professional—

Assembly Automation

A Management Handbook

Frank Riley

A thorough, complete guide to the WISE SELECTION and PROPER APPLICATION of commercially available equipment for the entire management team.

...with an excellent overview of considerations that virtually guarantee successful: specification, procurement, design, manufacture, installation, and sustained productivity of mechanized assembly systems. Everything from technical and economic feasibility through detailed operational procedures—to insure the best possible operation at lowest cost.

352 pp., illus. \$39.95

Mathematics, programming, and control

Robot Manipulators

Written by Richard P. Paul

Written by one of the world's leading authorities on robot manipulation, this is the first book in the field that literally covers all aspects of computer control of mechanical manipulators.

It is without a doubt the most complete work available on control of robot arms and manipulators and will be a guide and valuable reference work for many years to come.

Published by M.I.T. Press, 279 pages Price: \$79.95

Robot Motion: Planning and Control

Edited by Michael Brady, John Hollerbach, Timothy L. Johnson, Tomaz Lozano-Perez and Matthew T. Mason

This major work is geared to the development of robotics. In an effort to aid senior and graduate-level students in the field, the editors have grouped the book into five main chapters: Dynamics; Trajectory Planning; Feedback Control; Compliance and Force Control and Spatial planning.

What makes this book 'unique' is that each chapter is introduced by a substantial analytical survey that lays out the problems in that area of robotics and the approaches and solutions that have been tried, with an evaluation of their strengths and shortcomings. There is even an overall view in the development of artificial intelligence. This book is a "must" for anyone interested in the robotics field.

Published by M.I.T. Press, 585 pages Price: \$37.50

Tool and Manufacturing Engineers Handbook, Fourth Edition

Volume 1 Machining

Published by Society of Manufacturing Engineers

The Tool and Manufacturing Engineers Handbook, 4th Edition has arrived and it's jam packed with the data you need to meet the competitive demands of today's manufacturing environment. Volume 1 Machining, contains specific information vital to every company that makes chips in metal, plastic, and all other engineering materials.

Over 600 machining experts have contributed to the many special features of Volume 1 which include: careful attention to metric equivalents, up-to-the-minute information on computerized numerical control, an extensive machine controls glossary, concise set of essential machinability tables, and published in 15 years, plus a cross-referenced index of 7880 items. 1500 pages, 1983.

\$89.00

CAD/CAM, CAE Survey, Review and Buyers' Guide Comprehensive Guide

The CAD/CAM Survey and Buyers Guide is a reliable, cost-effective reference that details the features, capabilities and limitations of CAD/CAM systems and technologies. Designed to save you time and effort, this all-inclusive, easy-to-use Guidebook is filled with decision-making data about industry's vital yet least understood tool - CAD/CAM.

Written in clear, jargon-free language, indexed, and organized for easy reference, the Guide is a critical review of systems and technologies that show you how to prepare a justification and plan the acquisition of a system. 252 pages, 137 illustrations

\$199.00

Computers in Manufacturing Principles and Practices for Managers and Engineers in Metalworking

Edited by the Staff of American Machinist

This book was developed to meet the information needs of executives, managers, and engineers - both manufacturing and design - who are responsible for efficient performance and productivity in metal products industries. It presents a comprehensive overview of the computer's role in modern metalworking.

The book covers computer integrated manufacturing (CIM); design for products and tools; graphics; computer simulation for products, processes, and systems planning; procedures for justifying the investment in computer equipment; and management uses of computers - from mainframes to micros - for planning and control. The new concept of flexible manufacturing systems (FMS) is thoroughly discussed, and several chapters are devoted to specific applications and to case histories for metal cutting, metal forming, assembly, finishing, inspection, and robotics. 3 1/2 x 10 1/4 320 pages.

\$33.95

Systems and Technology for Advanced Manufacturing First Edition

Compiled by Society of Manufacturing Engineers

Change just one part of your production system and you create an imbalance-chaos. Use the information in Systems and Technology for Advanced Manufacturing to improve your systems with an integrated approach-and you create an increase in profit.

In your copy you will study the operational characteristics of many of the tools of advanced manufacturing technology: computer systems to generate process plans, manufacturing software, microcomputer simulation, automatic inspection systems, lasers, robots, and more.

Hardcover: \$32.00

Save time and money with this collection of 114 ready-to-run BASIC programs for the hobbyist and engineer.

BASIC COMPUTER PROGRAMS IN SCIENCE & ENGINEERING

By Jules H. Gilder

This book is packaged with two diskettes containing 114 BASIC programs for the hobbyist and engineer that can be used on the Apple II, II Plus, and IIe microcomputers.

The package provides ready-to-run programs for statistical and design applications. Programs include instructions for means; standard deviation averages; curve-fitting; interpolation; designing filters and attenuators; matching networks and histograms; matrix mathematics, and complex numbers. Each pre-tested program is listed in BASIC with a sample run.

256 pages, 6x9, two 5 1/4" diskettes for the Apple II, II Plus, and IIe, documentation.

\$34.95

Please send me the following management book(s). I understand that if I'm not satisfied for any reason, I may return book(s) in salable condition within 15 days for full refund.

- ☐ Robot Motion: Planning and Control.....\$39.50
- ☐ Robot Manipulators.....31.50
- ☐ An Introduction to Robot Technology.....29.95
- ☐ The 1984 S. Klein Directory of Computer Graphics Supplies.....60.00
- ☐ Computers in Manufacturing.....33.95
- ☐ Principles of Numerical Control.....24.95
- ☐ Industrial Robotics Handbook.....32.50
- ☐ Assembly Automation.....39.95
- ☐ Robotics Research: The First International Symposium.....65.00
- ☐ Systems and Technology for Advanced Manufacturing First Edition.....32.00
- ☐ CAD/CAM, CAE Survey, Review and Buyers' Guide.....199.00
- ☐ Basic Computer Programs in Science & Engineering.....34.95
- ☐ BASIC Programming Solution for Manufacturing.....32.00
- ☐ Tool and Manufacturing Engineers Handbook, Fourth Edition.....89.00

Please allow 4-6 weeks for delivery

New York Addresses add 8% sales tax

*Shipping Charges \$
Total \$

* Shipping Charges: Add \$2.50 for each book. For air service to Canada and Mexico add 30% of Book price; to Europe or South America add 35% to Asia, Africa, or Australia, add 40%

ADVANCE PAYMENT
REQUIRED -
CHECK OR
CREDIT CARD

MAIL TO: GRANT PUBLICATIONS, INC.
Book Division
160 East 48th Street
New York, N.Y. 10017 U.S.A.

I understand this has a 15 day money-back guarantee.

☐ Payment enclosed ☐ Credit card
For credit card billing, please check: ☐ VISA ☐ MasterCard

MasterCard Interbank No. Exp. Date

Account No.

SIGNATURE DATE

SHIP BOOKS TO:

Name Title

Company

Address

City State Zip

As an example, the text box shows how the strings "TOO" and "TWO" compare using this algorithm.

CHIP STRUCTURE

The PF474 is composed of three subsystems: DMA controller, proximity computer, and ranker.

The DMA (direct memory access) controller allows direct access to the contents of memory without going through the central processor. The DMA controller is operated by writing a starting address into a register in the PF474. The PF474 then requests control of the system bus and reads memory sequentially until a *null* character is read. The DMA controller can load up to two characters per microsecond.

The proximity computer subsystem is the heart of the PF474. This is where the string comparison algorithm is performed which indicates the similarity of two strings stored in string memory. Character parameters can be loaded into the proximity computer which will control the precise manner in which the similarity is computed. The proximity value is a 32-bit binary fraction ranging from zero to one. A value of one indicates that the words are identical. A value of zero indicates that the words are completely different. The exciting feature of the comparison is that it measures up very well with human intuition. The function "sees" the same similarities that people do.

The ranker is the third subsystem founded in the PF474. The ranker takes the results of the proximity computer and saves the best matches in the rank list memory. The proximity values and a four-byte internal record number for each of up to 16 best comparisons can be stored in the rank list memory. The internal record number is initialized by the user and is automatically incremented by the ranker subsystem after each comparison. This generates a unique internal record number for each comparison.

The proximity computer and the ranker *pipeline* data. This means that the proximity computer can be performing a new computation while the ranker is still ranking the previous computation. Since the ranker works concurrently with the proximity computer, computations are not slowed. The pipeline structure of proximity computer and ranker is the secret which allows the PF474 to perform comparisons at such high speeds.

Example Text Comparison

When the PF424 chip compares two strings, it computes a proximity value that is calculated by the following equation:

$$P.V. = \frac{2 \times (\text{A compared to B})}{(\text{A compared to A}) + (\text{B compared to B})}$$

As an example, the proximity value for the comparison of the strings "TOO" and "TWO" is shown here.

First, write the words above one another:

TOO
TWO

Look at the first column:

T
T

The letters are the same and give us the first match (match = 1).

Now look at the first two columns:

TO
TW

Compare the letters and find only one match (match = 1).

Now look at all three columns:

TOO
TWO

Comparing the letters results in two matches (match = 2).

Now look at the words with the letters reversed:

OOT
OWT

The first column comparison produces one match.

The first two column comparison produces one match.

The first three column comparison produces two matches.

The total number of matches found are eight.

Now you must calculate the value for "TOO" compared to itself.

The first column comparison produces one match.

The first two column comparison produces two matches.

The first three column comparison produces three matches.

The reverse matches are, of course, the same.

A total of 12 matches are found.

Comparing TWO with itself also results in a total of 12 matches.

Placing these values into the proximity value equation as such:

$$P.V. = \frac{2 \times 8}{12 + 12} = \frac{16}{24} = \frac{2}{3}$$

produces a proximity value of 0.66666666.

APPLICATIONS

The Proximity PF474 is useful not only for spelling checking—it is designed to artificially interpret any data input in an incorrect form, whether from a human operator or from a process controller. Here are some of the more interesting applications.

If the search space is kept to less than 1000 items, the PF474 responds almost instantly. This means that the chip can be used to make operating systems friendlier by responding correctly to incomplete or misspelled commands.

The medical and legal professions depend on searching for information in full text histories. Because of the searching speed of the PF474, such searches could be performed using microcomputers rather than high speed mainframes.

DBMS systems using the PF474 would operate in a real-time interactive mode rather than the inquire, wait, and try again mode. As the user typed the inquiry, the PF474 would be searching, comparing, and narrowing down the list being displayed.

The PF474 can search large databases as the information comes off the disk. Such a unit could be built into a hard disk peripheral and would function as a database engine.

One application is for speech recognition. Since the PF474 can handle strings of up to 127 characters, rather detailed phonetic representations can be handled. The PF474 could then compare the phonetic representation against a set of templates for a medium vocabulary speaker-independent recognition system.

Optical character recognition of both typed and handwritten characters is another suitable application of the PF474. Again, the representations would be compared against a set of templates. The system, of course, could be used in any visual recognition system.

The greatest operating speed in any application will be obtained if more than one PF474 is used. Each PF474 can be connected in parallel and can use the same query string on different parts of the data to be searched. While most applications

will have static searching parameters, some applications require parameters to be dynamically modified. Since the PF474 holds the parameters in programmable memory, they can be dynamically modified in adaptive pattern-recognition applications.

Proximity Technology, Inc. has also developed a standard interface board for the IBM PC which uses the PF474 chip. The board is available with diagnostic and demonstration software. Evaluation chips can be purchased for \$250, and the IBM PC board costs \$1295. For further information please contact Lori Fratilla at Proximity Technology, Inc., 3511 NE 22nd Ave., Fort Lauderdale, FL 33308, telephone (305) 566-3511.

Reader Feedback

To rate this article, circle the appropriate number on the Reader Service card.

73	83	93
Excellent	Good	Fair

HOBBY ROBOT-X™

SEQUENCER BOARD Allows generation and playback of controlled sequences without a computer. Up to a four-minute controlled sequence or selectable shorter sequences. Can control Mobile Mouse and Robo Rat directly or used to control automation kits. Can also be used on board Logi-Cat. KIT \$25. Up grades available with more memory including analog functions.

RS-232 COMPUTER INTERFACE For experiments with Radio Shack, IBM, etc. KIT \$25

APPLE PORT For experiment with Apple II computer KIT \$25.

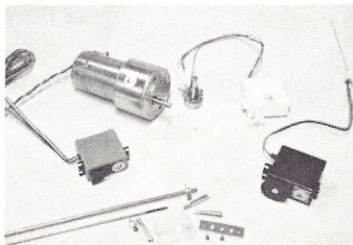
EXPERIMENTERS

Our affordable hardware allows hands on experience with advanced technology in all areas of ROBOTICS and INTELLIGENT MACHINES. If you want to do more than read about it, try at least our ELEMENTARY AUTOMATION KIT. \$25.

INTERMEDIATE AUTOMATION KIT With servo and interface circuit, A/D circuit and much more. KIT \$50.

COMBINATOR EXPERIMENTERS KIT. Has INT. + ELEM. + BUILDERS KIT. A/D, DAC temp, pos, force sensors. KIT \$100.

COMPONENTS FOR SPECIAL AUTOMATION servos, sensors, interface, etc.



SCIENCE FAIR PROJECT IDEAS

A booklet with timely research ideas for science fair projects. Credible and relevant research topics which either use the power of computer automated data acquisition as a tool or involve machine vision and intelligence. Order booklet alone for \$10 ppd. With AUTOMATION KIT \$5. Included with EXPERIMENTERS KIT.

EXTRA LOW COST ROBOTS

Science fair—Christmas presents and expanded group of economy robots for science fair projects and robotic beginners. Excellent for kids. Expandable into very versatile systems.

KITTEN MINI ROVER A smaller, 6" system, includes same 7 channel RC, motor controller and computer interface. KIT \$200. ASSY \$330.

MINI ARM (the cats paw) adds pick up action. KIT \$60. ASSY \$100.

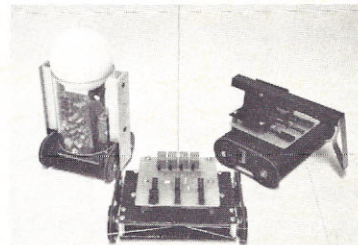
RUG RAT (R2 micro rover) slightly smaller. KIT \$200. ASSY \$330.

NAVIGATION AND TRACKING INQUIRE

LOGI-CAT (same size as Kitten Mini Rover) The independent cat. A self-contained version of our kitten mobile robot provides a versatile robotics lab without using a computer. Can be wired to track or avoid light, follow lines, avoid objects, and more. KIT \$75

MOBILE MOUSE Our lowest cost mobile robot is self-controlled by flexible wires from a computer interface or sequencer board. Can draw on paper, detect obstacles, turn, pivot, run backward or forward. KIT \$50

ROBO RAT (R2 style of above) KIT \$60



DEALERS WANTED

Advanced systems available including INDUSTRIAL VISION SYSTEMS.
WRITE OR CALL FOR CATALOG



SPECTRON INSTRUMENT

1342 W. Cedar Ave. Denver, CO 80223
Telephone (303) 744-7088



A BIOLOGICAL MODEL FOR MULTILEGGED ROBOT CONTROL

Thomas A. Easton
Box 705, RFD 2
Belfast, ME 04915

In my last article, "Bipedal Balance," (*Robotics Age*, April 1984, p. 21) I described a concept called a Reflex Autonomous Machine. The mechanism mimics the reflexes an animal uses to preserve balance. These reflexes were the vestibular, placing, flexion, and stretch reflexes. They worked by automatically transforming sensor readings into appropriate corrective movements of a robot's legs.

My focus was on bipedal animals such as humans and on bipedal robots. However, these same reflexes serve in quadrupedal animals. Since I am discussing legged robot "animals," I could not resist playing the acronym game. My concept for a multilegged robot steering mechanism is called HORNS, a Heuristically Orienting Navigation System. It is designed to reduce the robot's need to use a central control computer to do much more than make very general decisions such as "Keep watching that," "Investigate that," or "Carry on." The HORNS will take care of the steering.

HORNS' underlying principle seems basic to animal steering: the body follows the head. You can see this principle in action very nicely in the praying mantis. When a mantis sees an insect off to one side, its eyes are not stimulated equally. The eye closer to the prey records a larger, brighter image than the other. The head rotates until the eyes' images are equal and the head's axis of symmetry is aimed at the

prey. The body then swings around behind the head to bring the forelimb grabbers in-to line with the prey. This reflex sequence is controlled by signals from a clump of bristles on each of the mantis' shoulders which are disturbed or bent as the head moves. When head and body are in line, the bristles on the two shoulders are equally disturbed, their signals show no difference, and the forelimbs are triggered to seize the prey. All is automatic. The mantis' puny brain need only say, "Looky there!" and all else follows.

Mammals such as cats, dogs, and humans follow the same principle in another way. They use the *tonic neck reflexes* to generate changes in limb con-

figuration in response to head and neck movement. As the head bends downward, toward the chest, sensors in the neck respond and the forelimbs flex while the hindlimbs extend. If the head bends upward, the forelimbs extend and the hindlimbs flex. The effect is to line the body up with the head as the head dips to examine some object or to facilitate movement on a grade. If the head turns to one side, the limbs on the side toward which the jaw points (the *jaw-side*) both extend and the others flex, as shown in Figure 1. These actions are also evidenced when a baseball player goes after a fly ball or a fencer takes the en garde position.

The lateral tonic neck reflex is the one most useful in steering. It need not actually produce limb movement, for it is generally conspicuous only under special conditions. But even in humans, it changes the tone of the limb muscles and aids the motor neurons, the final common path, so that willed movements following the pattern of the reflex are easier to perform. In fact, the reflex may be best interpreted as a *readiness* to move in a particular way. It steers because a reflex-stiffened (jaw-side) limb will tend to function as a pivot around which the opposite limb can step.

Figure 2, which shows a dog getting to its feet and turning, demonstrates this steering effect. As the dog rises and its head begins to turn, the jaw-side limb ex-

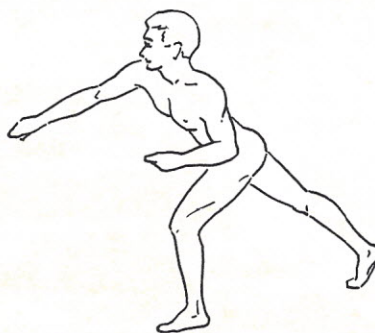


Figure 1. The pattern of the tonic neck reflex in a man who has just completed throwing the javelin. Traced from Plate 48 of Eadweard Muybridge's *The Human Figure in Motion* (Dover, 1955).

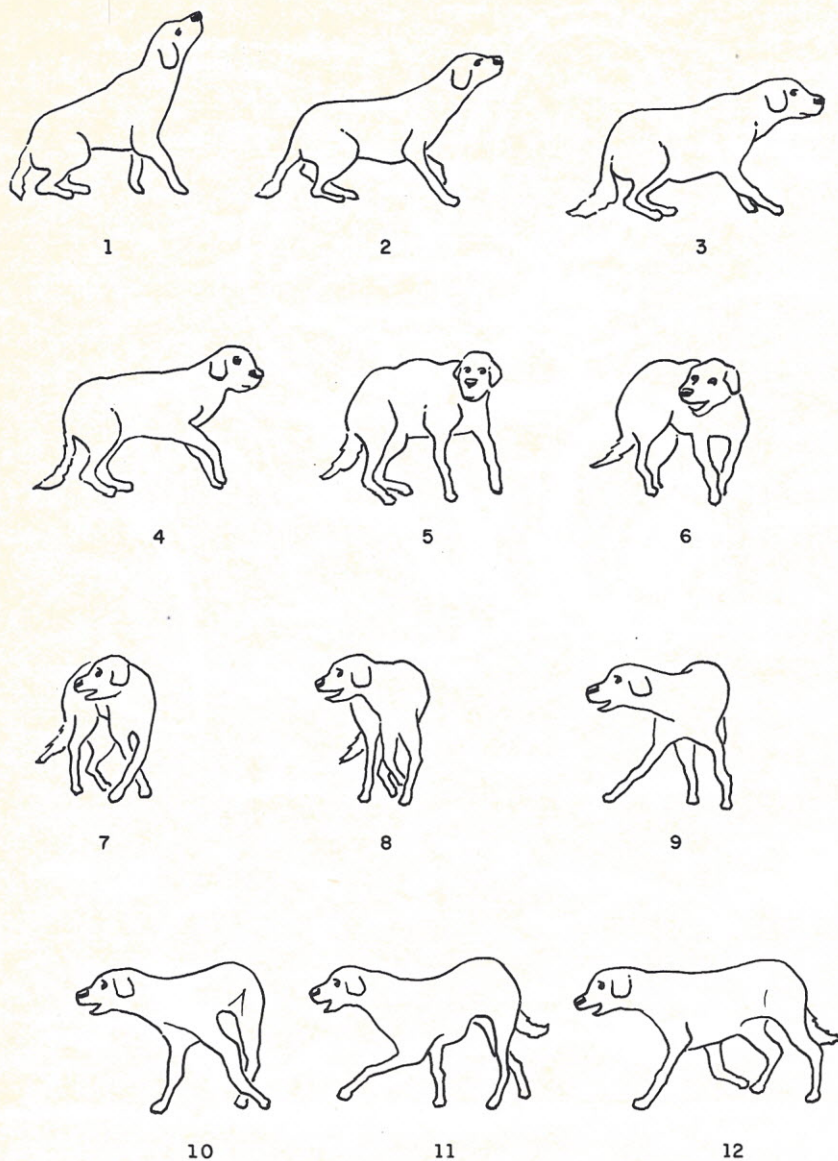


Figure 2. Twelve states (0.174 second apart) in the rising and turning of a dog. These stages show the interaction of head and limb position. Traced from Plate 123 of Eadweard Muybridge's *Animals in Motion* (Dover, 1957).

tends. Then, starting in frame 5, that limb serves as a more or less rigid pivot about which the animal steps. As the dog completes its turn and the head begins to straighten out on the body (in effect moving toward the opposite side) the limb roles reverse. The tonic neck reflex seems to unbalance ordinary linear locomotion in such a way as to make a turn easier to perform in the direction indicated by the head's orientation.

The tonic neck reflex works similarly in humans. As the head turns, the jaw-side leg stiffens while the other steps around it, and the body turns to follow the head. Sometimes, however, the reflex can cause problems. A bicyclist or car driver knows how a sudden turn of the head can cause a swerve—largely because of the reflex ef-

fect on the limbs. Humans must sometimes deliberately overrule their reflexes. Fortunately, they can.

Clearly, the lateral tonic neck reflex could be very useful to a robot. Whenever it directs its instruments or eyes toward a particular object or direction, the body would follow along. To turn left or right, it would only need to look left or right. The turn would follow as the reflex appropriately biased the standard, straight-line locomotor program.

The nonlateral tonic neck reflexes can play other roles. Suppose that a quadrupedal robot is walking toward some object on the ground. As it approaches, it lowers its head to keep the object in view. As the head bends downward, the tonic neck reflex flexes the forelimbs, bringing

the eyes nearer to the object, while it stiffens the hindlimbs, stopping movement. If, as the robot approaches, the object flies off into the sky, the head bends upward, the hindlimbs flex, and the forelimbs stiffen. Again the machine stops—and sits down, staring doggishly after its elusive goal.

As long as we rule out some external controller, our next interesting question is then, "What makes the head move?" For animals, there are several answers: The head moves, or is prepared to move, in accord with eye movements, gravitational effects on the otoliths (linear accelerometers in the inner ear), and movement effects on the semicircular canals (angular accelerometers in the inner ear).

The otolith effects are simplest. When the head is not level, the pull of gravity on the otolith organs produces signals which bring the head back to that orientation. If the neck is kept from bending, the limbs flex and extend to level the entire body. If the neck is free to bend, the otolith effects then take care of leveling the head on the neck, and the limbs and body then follow the tonic neck reflexes.

You can see this interaction of otolith and tonic neck reflexes in the righting reflexes. If you lay a horse on its side with its head flat on the ground, the otolith response first brings the head upright, the upper ear against the upper shoulder. This evokes a tonic neck reflex that extends the upper limbs and flexes the lower. As the otolith reflex then levels the head to near the normal position, the trunk twists and a second tonic neck reflex extends the lower limbs and brings the animal to its feet. The sequence is coordinated at the lowest level of the brain, and (at least in the horse), is obligatory. The initial head movement must occur if the animal is to get to its feet at all. If you prevent the initial head movement by holding the head down, the animal cannot rise.

The semicircular canals and their responses to angular acceleration are very useful in maintaining balance, but they can also help steer. In this context, we need only consider horizontal acceleration such as an animal would encounter on a turntable or during a turn. The reflex response to such movements is a repetitive oscillatory movement of the eyes, head, and ears called *nystagmus*. It consists of a fast flick in the direction of rotation, followed by a slow drift in the opposite direction. This

reflex serves to prevent apparent movement of the environment by maintaining sensory fixation on environmental features. Nystagmus also occurs in the laboratory when an experimenter rotates the environment around an animal and in nature when an animal watches a stream of movement such as a brook, drifting clouds, or a line of other animals. Here, though, it is not a reflex response to fluid moving in the semicircular canals. Instead, it is an automatic product of the brain's attempt to fix on a moving stimulus. If the stimulus is visual, the eyes and head will track it; they will also track sounds and odors. The result is a *sensorikineti*c nystagmus that keeps the head aimed at the stimulus. Tonic neck reflexes ensure that the animal is always ready to pursue the moving stimulus, a readiness of obvious utility to any predator—or to a tennis player.

Still other reflexes may link eye movements to head movements and thence body movements. The eye muscles have stretch receptors much like those in other muscles that serve the stretch reflex. When the eyes move, certain of the eye muscles are stretched, and the stretch receptors generate signals that can affect the neck and limb muscles. The effect of this reflex seems to be to prepare the body to move away from whatever has attracted the gaze. Perhaps it is an avoidance mechanism, reflecting a basic evolutionary axiom—those animals whose first reaction to anything that attracts their attention is fear and readiness to flee may have the best chances of surviving. On the other hand, since a turn of the head must evoke a nystagmic movement of the eyes in the opposite direction, it could be a means of reinforcing the necessary coordination of a turn.

A robot might not use a tonic eye reflex in quite the animal's way, since it certainly has no great need to be ready to flee. Thus, a robot's eye reflex might work just like the lateral tonic neck reflex. This may seem redundant, but movable eyes offer a low-mass, quick-response, low-energy way to prime the body for a turn. They also tie steering a bit more directly to visual search, fixation, and decision making. It also ties steering to other distance senses. When you hear a noise in the dark, your eyes promptly move toward the source of the noise. Your eyes can serve as an intermediary between hearing—or other senses—and steering.

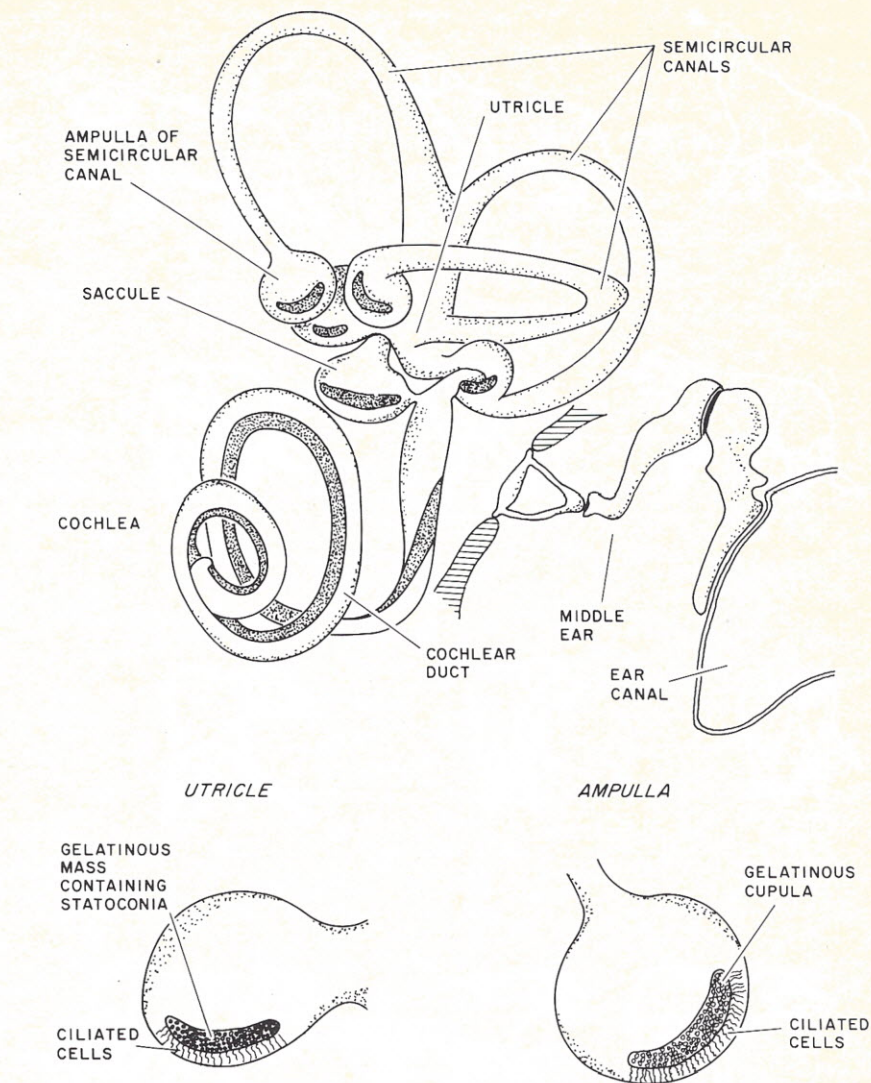


Figure 3. The vestibular apparatus consists of the semicircular canals, the utricle, and the saccule. All are contained in the inner ear.

All the reflexes we have discussed are imitable. Yet it might be wasteful to try to equip a robot with all of them. After all, a robot is more simply designed than a cat or a human being and its sensors may not move independently. Let us, then, picture a bipedal or quadrupedal robot whose head can turn, but whose distance sensors (for vision and sound) are mounted rigidly on the head. It seems fairly straightforward to design the robot's circuitry so that when it turns its head to fixate on some stimulus, reflexes like the tonic neck reflexes prepare its limbs to move toward the stimulus as soon as the central computer decides to move. An intermediary *tonic eye reflex* would be necessary only if the sensors could move in or on the head. This reflex might be necessary for large robots equipped with rotating radar scanners or the like.

Tying such reflexes into the balance system could provide a very useful way to get

a fallen robot back on its feet. As in the horse, there need only be an otolith-like orientation sensor in the head which triggers reflexes to keep the head level. As the head moves, it will in turn trigger tonic neck reflexes that will bring the body to its feet.

The HORNS concept models animal reflexes to create a robot that can steer itself automatically. The model simplifies the problem of robot motor control to its essentials. The ultimate control becomes simply attentiveness, so that any decision to move will move the robot toward whatever holds its attention. It requires only that the robot have suitable sensors for joint or eye movement, plus circuitry that allows signals from these sensors to affect appropriate motor control centers. The result is that the central computer or outside controller has much less controlling to do. In fact, the central computer need do little more than control where the sen-

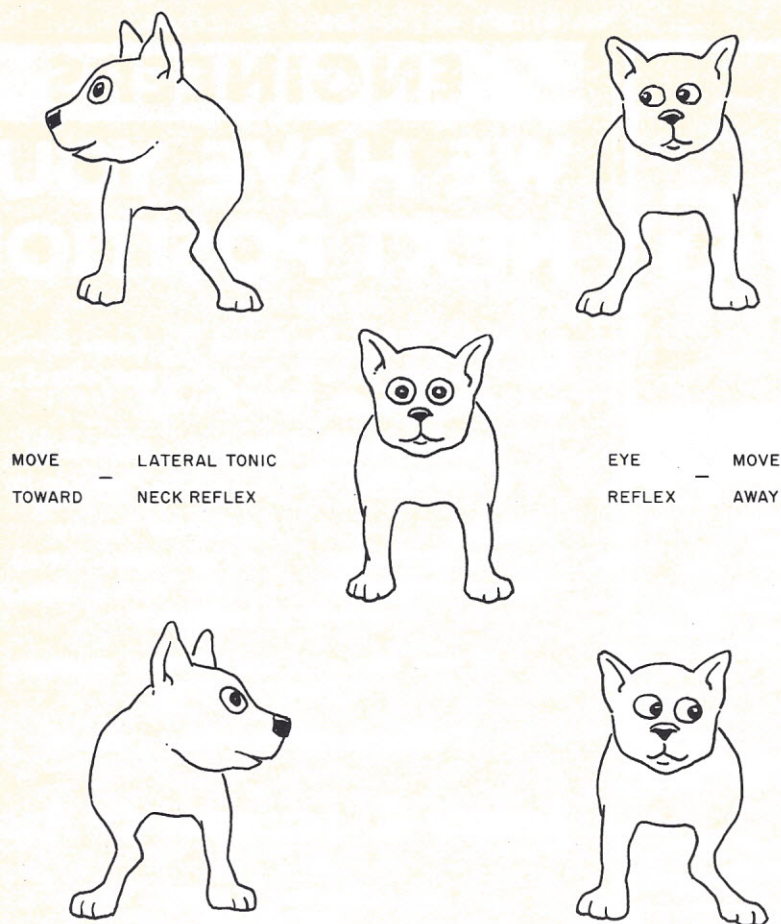


Figure 4. Neck and eye reflexes prepare the body to move in opposite directions.

sors point and say, "Go!" Reflexes take care of all the rest.

Can we build a Reflex Autonomous Machine with HORNS? The basic design, worked out by eons of evolution, is a proven success. But is our technology up to duplicating it? We need legged bodies. We need small, cheap microcomputers to control the body parts, perhaps one for each limb. We need sensors for joint movement, acceleration, and gravity. We have all these. What we don't have yet is the programming that will make all these parts work together as smoothly as an animal's parts. But it need not be long before we have a robot with just that programming, one that can be turned loose to explore the moon or Mars or the bottom of the sea with a minimum of guidance. It will steer itself, mostly stay upright, and rise to its feet if it falls. It will truly be an autonomous machine.

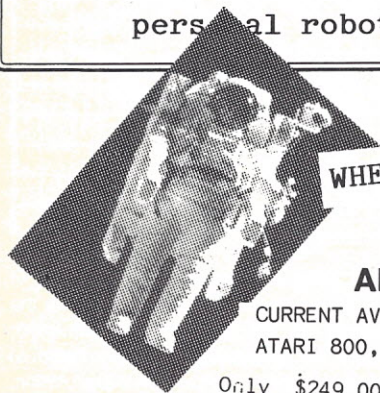
Reader Feedback

To rate this article, circle the appropriate number on the Reader Service card.

74	84	94
Excellent	Good	Fair

AMAROBOT

personal robot store

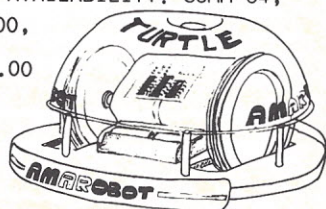


WHERE THE FUTURE IS

PRESENTING THE AMAROBOT TURTLE

CURRENT AVAILABILITY: COMM-64,
ATARI 800,

Only \$249.00



BASIC TURTLE: software included
Programmable, Dual motor,
Independent, Variable speed
2 I/O, Cable, Paddle drive-teach
4 prog. LEDs, Wheel encoder
Pulse width modulated

OPTIONS NOW AVAILABLE:

Vacuum attachment: \$35.00
Forth language: \$85.00
2 sets bumper sw. pr. \$19.95
32 outputs attachment: \$49.95

OPTIONS SOON AVAILABLE:

Arm, Body, Voice, Vision,
Logo Language, Graphics
Infrared data link, RS232,
Real time clock,
Course manuals. Kit forms,
Hall encoding, Mind control

AMAROBOT

Richard Amoroso
(415) 548-3554

1780 Shattuck Ave.
Berkeley, CA 94709



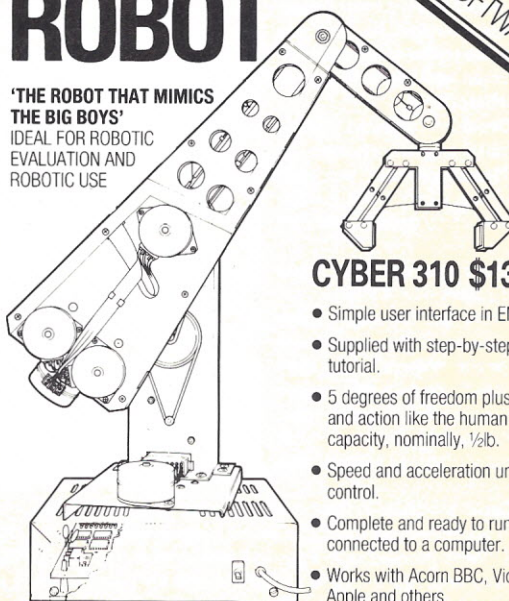
Catalogue \$3.00
refundable



Circle 1

THE FORTH ROBOT

'THE ROBOT THAT MIMICS
THE BIG BOYS'
IDEAL FOR ROBOTIC
EVALUATION AND
ROBOTIC USE



CYBER 310 \$1399

- Simple user interface in ENGLISH.
- Supplied with step-by-step operating tutorial.
- 5 degrees of freedom plus grip. Size and action like the human arm. Lift capacity, nominally, 1/2 lb.
- Speed and acceleration under user control.
- Complete and ready to run when connected to a computer.
- Works with Acorn BBC, Vic 64, Apple and others.

Details from

Acorn Computers Corporation
400 Unicorn Park Drive, Woburn,
Massachusetts 01801
(617)-935-1190



AVAILABLE WITH
XYZ, CARTESIAN AND
POLAR SOFTWARE

FREE CATALOG

ROBOTICS
ARTIFICIAL
INTELLIGENCE

BOOKS

If you've been looking for books on Robotics and Artificial Intelligence, but can't find them at your local bookstore, try us. We specialize in Robotics/AI books for hobbyists and professionals. Whether you're interested in building a robot, the state-of-the-art in industrial robotics, or intelligent machines, we have the selection and service you won't find elsewhere. Join our hundreds of satisfied customers. Circle our reader service card number or send this coupon directly to us for fast response.

TECH BOOKS

P.O. Box 16265 Alexandria, VA 22302

TECH BOOKS P.O. BOX 16265 ALEXANDRIA, VA 22302

Send a catalog to:

Name _____
Address _____
City/State _____
ZIP _____

Your Source for Robotics/AI Books

RA1184

INTRODUCING THE AFFORDABLE ROBOT

Have you avoided joining the robot revolution because the robots you have seen were either too toy-like or too expensive? Wait no longer. HoB-Bots of Hudson introduces **MicroDroid**.

MicroDroid has advanced features usually available only on more expensive robots. **MicroDroid** has two independently-driven wheels which move its rugged all-metal body through the thickest carpets.

MicroDroid has a hand with a 3" span capable of holding over 4.5 lbs.

MicroDroid has five major sensor systems which permit it to interact with its environment. Within the hand there are two variable pressure sensors, two hand-limit sensors, and an optical object sensor. **MicroDroid** also has a tactile sensing system and an optical shaft encoder for position sensing.

MicroDroid is available in three models to meet your specific needs. Model MDI00C is supplied with our robot development computer programmable in BASIC. Model MDI01P is provided with a parallel interface to your computer. Model MDI02Z is provided with an interface which plugs into the expansion bus of your Timex-Sinclair computer.

Now you don't need to wait any longer. Now you can join the robot revolution with **MicroDroid** and HoB-Bots of Hudson.

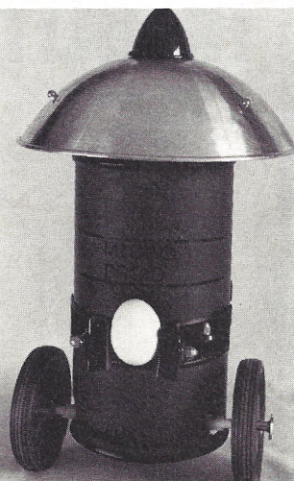
HoB-Bots of Hudson

Robots for Home and Business

P.O. Box 460

Hudson, Ohio 44236

(216) 656-3413



Model MDI00C \$595
Model MDI01P \$495
Model MDI02Z \$545

ENGINEERS WE HAVE YOUR NEXT POSITION

Engineering & Scientific Career Placement is our only business, so you know you're getting specialized service and top opportunities NATIONWIDE.

DESIGN • DEVELOPMENT • RESEARCH
• MANUFACTURING Positions Available In:

- Robotics
- Sensor Technology
- Factory Automation
- Control Systems
- Hardware/Software
- Mini/Micro Computers
- Artificial Intelligence

All fees, relocation and interview expenses
COMPANY PAID.

Contact us with or without a resume.

James E. Iannoni & Assoc.

P.O. Box 128
Canterbury, CT 06331
203-546-9448

PERSONAL ROBOTICS MAGAZINE

The first magazine of its kind, PRM pioneers the exciting field of personal robotics, an industry certain to revolutionize our lives as the computer did only a few years ago. You, the personal robot enthusiast, user, and/or developer, now have the opportunity to receive Personal Robotics Magazine, the publication designed to grow with the revolution.

Yes! I want to subscribe to PRM.

NAME _____

STREET ADDRESS _____

CITY _____

STATE _____

ZIP CODE _____

Please send me:

- ☐ 1 year (6 issues) of PRM. (\$21.)
☐ 2 years (12 issues) of PRM. (\$40.)
☐ Bill me. ☐ Check enclosed.

Send coupon to: PRM-Subscription Dept.
P.O. Box 421
Rheem Valley, CA 94570

- Helpful information on personal robots and their accessories.
- Programs and projects for: Androbots, RB5X & Hero-1.
- Forum for personal robot enthusiasts, users, and hobbyists.
- Users' groups listings.
- Inside information on robot companies and their latest developments.
- Insightful essays on the future of robots.
- Plus much, much more!

INTRODUCING A SINGLE-BOARD COMPUTER YOU CAN'T PASS UP INTRODUCING THE ZINGER COMPUTER SYSTEM

Have you ever needed a small, low power
easily programmable application board?

This is it!

Zinger CPU • 8073 CPU with Tiny BASIC • 2K x 8 RAM expandable to 12K on board • Board supports JEDEC family ROM/RAM • EEPROM programmer on board • EEPROM/RAM interchangeable • 24 line programmable parallel I/O port • Optional on-board switching power supply supplies +12 & -12V for all interface devices so that only +5V 300mA need be supplied • All devices memory mapped for ease of programming •

ASSEMBLED & TESTED \$114.95

ZINGER INTERFACE BOARD

Board may be purchased with any or all combinations • 276 word speech synthesizer • 8 channel A/D converter • RTC with programmable latches & battery backup • programmable keyboard interface with 64 keys or sense switches for remote sensing • 8 char alphanumeric display or intelligent 16 char by 2 line LCD display • 24 line programmable parallel I/O port • all devices connected to interrupt register • all devices memory mapped •

UNIVERSAL AC/DC CONTROL BOARD

Controls 14 6A 200V optically isolated devices & 10 transistors for high current switching • 32 sense or drivelines • all sense/drive lines use CMOS bilateral switches so that it may be used for an 8 to 1 line 0 to 5V analog multiplexer • board designed to use one 24 line PIO port or CPU or interface board •

AND FOR HOME CONTROL...

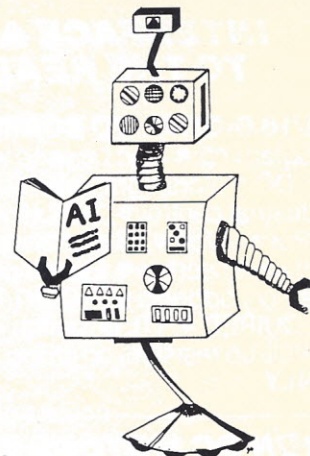
BSR INTERFACE FOR REMOTE CONTROL OF AC DEVICES

Compatible with BSR & other wall modules sold by Radio Shack, Leviton & others • driven from a parallel port •

SCHULZ ENTERPRISES INC.

1285 LAS TUNAS DR. SAN GABRIEL, CA 91776 (818) 287-5076

Hard to
keep up
with what's
going on
in AI?



You need

**Artificial Intelligence:
Bibliographic Summaries of the Select Literature.**

560 pgs. \$145.00 ppd.

the Report Store

910 Massachusetts Suite 503RA • Lawrence, Kansas • 66044 • USA • 913/842-7348

Yes, please send me _____ copy(s) of **Artificial Intelligence:
Bibliographic Summaries of the Select Literature.**

PAYMENT MUST ACCOMPANY ORDER

Name _____

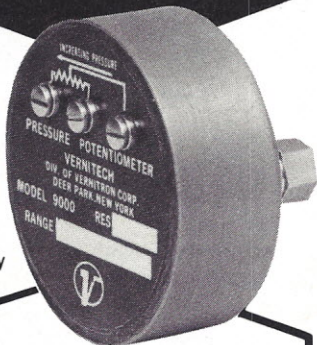
Address _____

City _____ State _____ Zip _____

INTRODUCING 2 NEW MODELS

#8000: 0 to 15, 30, 50, 75 psig
#9000: 0 to 100, 500, 1000, 5000 psig

Prototypes
immediately
available at
OEM quantity
prices.



LOW COST POTENTIOMETRIC PRESSURE TRANSDUCER GAUGE TYPE

SPECIAL FEATURES:

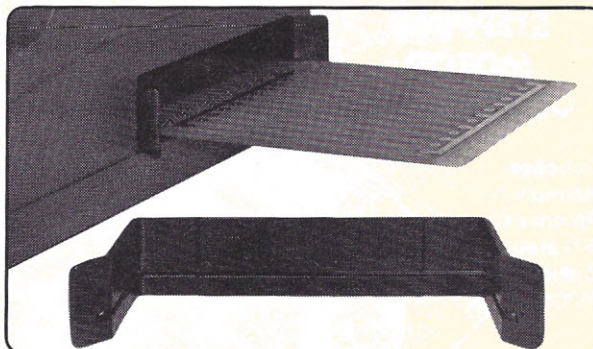
- HIGH VOLTAGE OUTPUT AC OR DC
 - REQUIRES NO AUXILIARY EQUIPMENT
- FOR: Signal conditioning, Amplifying,
Converting electrical inputs or outputs.



VERNITECH

a division of Vernitron Corporation
300 Marcus Boulevard, Deer Park, N.Y. 11729
(516) 586-5100 / TWX 510-227-6079

6809 SYSTEM DEVELOPMENT



EXPANSION HARDWARE FOR THE TRS-80 COLOR COMPUTER

XPNDRI™

SuperGuide™

CoCo Expander Card

Gold edge connector plugs into the CoCo cartridge connector. Signals are labeled on the bottom (wire side) with ground and power buses; plated through holes. The 4.3 x 6.2 inch glass/epoxy card is drilled for ICs and components. The finest bare breadboard for your CoCo. Includes 8 page **Application Notes** to help you get started.

Precision molded plastic insert designed specifically to align and support printed circuit cards in the CoCo cartridge slot; an unbreakable removable card guide. Patent Pending.

\$3.95 each

Available now from:

ROBOTIC MICROSYSTEMS

\$19.95 each or 2 for \$36

BOX 30807 SEATTLE, WA 98103

INTERFACE APPLE II/e® TO THE REAL WORLD

24/16 DIGITAL I/O BOARD • Easy programming • Can be used with board below to run DC motors, lights, alarms • Robotics, industrial control • 16 buffered digital outputs • Poke on/off • 16 digital inputs • No setup software needed • Each input has its own memory address so NO BIT TESTING IS REQUIRED • TTL or switch inputs • 1 byte input • Pull-up resistors on all inputs
ONLY \$89.00

RUN DC MOTORS BY COMPUTER

DC CONTROLLER BOARD • Run two permanent magnet DC motors in FWD/REVERSE/OFF from ANY 4 latched TTL outputs • Use with Apple II/e game I/O & 16 pin cable • Or use board above • 24 volt/5 Amp capacity • TTL isolated from motors • Protection logic
ONLY \$69.00

CALL (312) 530-4110 24 Hrs—7 days a week

CONTROL WARE, DEPT. R-3
P.O. Box 1467
Oak Brook, Illinois 60521

WE PAY SHIPPING! Non-local orders sent 2nd DAY AIR.
TERMS: Check, M.O. or COD. IL customers add 6 1/4%.

Acrobunch and Legioss Variable

Plastic Model Kits
\$17.00 each



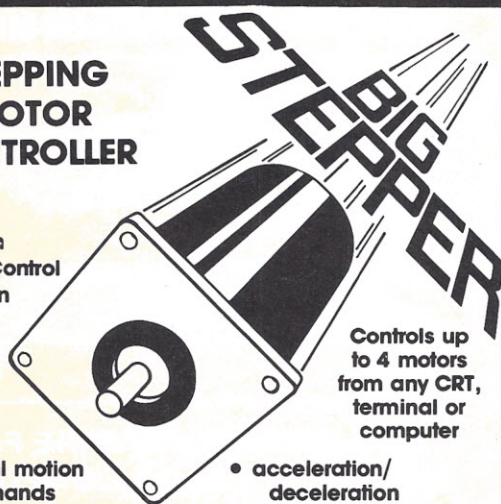
Robot Toys • Robot Plastic
Model Kits • Robot Books •
and More

Write For My Free Catalog

Space Robot Market
800 Bowie Lane
Greenwood, Mississippi 38930

STEPPING MOTOR CONTROLLER

Robotics
Animation
Process Control
3-D Motion
Drilling
Milling



Controls up
to 4 motors
from any CRT,
terminal or
computer

- powerful motion commands
- provides required power
- RS232 interface
- acceleration/deceleration
- simultaneous 4 axis moves
- senses 8 limit switches

STEPPING MOTOR TIPS COOKBOOK \$8
STEPPING MOTORS - priced from \$59
BIG STEPPER - Standard Version \$935
 - with on-board BASIC language \$985
 - with on-board CNC commands \$1950

Complete drive package for Apple II/IIe \$265
includes 4 motor positioning software

CENTROID (814) 237-4535
Box 739, State College, PA 16804



Take Control

Introducing the CYBOT Gripper. It is dc motor driven and is quiet and quick acting. At only \$249.95 it's ideal for education, research, and light-duty industrial applications. The Gripper features four sensory outputs: forward and reverse limit switches, jaw position and force. Additional specifications include:

- 0" - 2" grip range
- size: 1" x 6" x 3"
- 10 lb squeeze force
- replaceable jaw pads
- weight: 6 oz

A universal attachment plate, with through tapped holes, makes mounting the gripper easy.

Ordering Information

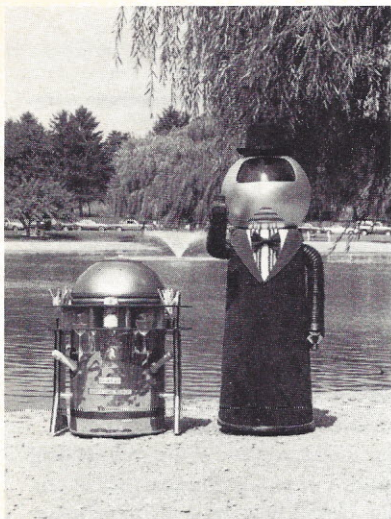
CYBOT Gripper \$249.95* (call for qty discounts)

(206) 827-6199

CYBOT, Inc.
733 7th Avenue
Kirkland, WA 98033

*Terms: P.O. or check. Washington residents add 7.9% sales tax.

ROBOTS



PROMOTIONS

TRADE SHOWS

VIDEOS

CONCERTS

PARTIES

Robots can be used effectively to attract attention and relate a high-tech image. They add fun and excitement to any function.

Digi-Tech robots are available for purchase or lease in many different styles and capabilities. They have appeared throughout the world, on television, at trade shows, in films, and at night clubs.

The new PARTY-TIME robots are equipped to serve drinks and entertain. Purchase one of these unique mobile bars for your next dynamic party.

For the ultimate in entertainment robots, contact:

Digi-Tech, Inc. (201) 548-4260
58 Van Buren Avenue, Metuchen, NJ 08840 USA

PERSONAL ROBOTICS EXCLUSIVELY FACTORY AUTHORIZED DEALERS

SENSATIONAL — CONTROL SYSTEM FOR HERO-1

Write, debug, edit, download programs from your Z-100, H-89, Apple II, CoCo, PC, others
MENOS System, includes RS232, 34K, OS in ROM
from Virtual Devices, Inc. List \$595.00 SALE \$540.00

BURN YOUR OWN RB5X PROMS

Introducing our Tiny BASIC PROM Development System

Hardware, Software, Documentation — Complete Package for 8073 CPU
VIC-20 Version \$299.95 Commodore 64 Version \$349.95
Dealer Inquiries Invited

ROBOTS	List	SALE	KITS	List	SALE
ANDROBOT			MOVIT		
Slave to your computer			Screwdriver Kits—No Solder—FUN		
TOPO II	1595.00	1485.00	11 Different \$23.65-\$70.15		CALL
Accessories		CALL	ROBOT SHOP		
COMRO			4 Different \$122.00-\$360.00		CALL
On-board 8-bit computer			Many Accessories		CALL
TOT		CALL	SPECTRON INSTRUMENTS		
Accessories		CALL	Many Different \$116.50-\$372.00		CALL
HUBOT			RHINO		
On-board 8-bit computer			Scorpion & XR-2		CALL
Premier 1000	3495.00	3275.00	Q-BOTS		
Accessories		CALL	Variable Configuration \$499.00-\$462.00		
IOWA PRECISION			2ND SOURCE ACCES.		
On-board 16-bit computer			Many Different for HERO-1, RB5X,		
Marvin MKI	5995.00	5785.00	Others		CALL
PERSONAL ROBOTICS CORP.			SOFTWARE		
On-board 8-bit computer			RB ROBOT		
Ropet-HR	2499.00	2210.00	Robot Control Language With SAVVY		
Ropet-XR	2199.00	2030.00	Apple 1 Disk	455.00	408.00
Accessories		CALL	Apple 2 Disk	595.00	533.00
RB ROBOT			RIO GRANDE ROBOTICS		
Download from your computer			Application PROMS For RB5X		CALL
RB5X	2295.00	2058.00			
Accessories		CALL			

TECHNICAL INFO—WE CAN HELP. CALL



Shipping Over \$200 Add 4%. \$200 And Under Add 5%.
Cash With Order Deduct 3%. N.M. Orders Add 4% Sales Tax.
Allow 4 Weeks For Delivery.



RIO GRANDE ROBOTICS RG

1595 W. Picacho #28, Las Cruces, N.M. 88005, Tel. (505) 524-9480

TEMPERATURE
VOLTAGE
PRESSURE
POSITION
AMPLITUDE
FREQUENCY
LEVEL
ANGLE
INTENSITY
SPEED

8 CHANNEL, 8 BIT ANALOG TO
DIGITAL CONVERTER WITH
RS 232C INTERFACE.

\$249.00

YOUR SENSORS AND THE SOFTWARE SCIENCE
MICROTELEMETRY MODULE CAN PUT 8 CHANNELS OF ANALOG
DATA AT YOUR COMPUTER OR TERMINAL'S FINGERTIPS.

SIMPLE ASCII COMMANDS CONTROL THE
MODULE.

RAW DATA, VOLTAGE OR SOFTWARE SCIENCE'S EXCLUSIVE
"TEMPACHEAP" LINEARIZED TEMPERATURE OUTPUT.

SENSORS, POWER SUPPLIES AND ACCESSORIES ARE ALSO
AVAILABLE. CALL OR WRITE FOR CATALOG AND APPLICATION
NOTES.

SOFTWARE SCIENCE (513) 561-2060
P.O. Box 44232, CINCINNATI, OHIO 45244



pneumatic valves
pressure controllers
flow controllers

MODEL	DESCRIPTION	PERFORMANCE	FEATURES
10	Electronic flow controller	0.003-0.7 SCFM	fast, linear, repeatable
12	Shaft-operated absolute pressure controller	0.5-32" Hga	control to 0.001" Hga
13	Shaft-operated differential pressure controller	0-110" Hgd	control to 0.001" Hgd
60	Proportional electronically controlled valve	0.5-110" Hg	control to 0.001" Hg
60N	Proportional electronically controlled low-noise valve	0.3-110" Hg	control to 0.0003" Hg
90	Voltage-actuated pressure controller	0-100 psi	absolute or differential

custom ranges available

SCHWIEN AND SON

293 Ventnor Avenue Aspen, Colorado 81611 303/925-5670

New Products

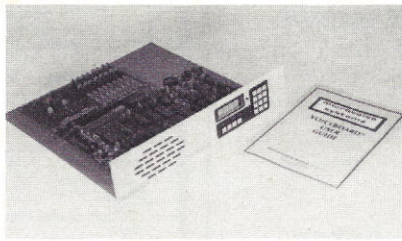
Protalker

The Protalker can provide any S-100, IBM PC, or IBM PC-compatible computer with voice output capability. A digital recording technique is used in the Protalker that easily facilitates any type of sound reproduction. Protalker can record and play back high-fidelity reproductions of voice, music, and sound effects. The Protalker provides natural-sounding reproductions of any speaker's voice and intonation in any language. The Protalker provides switch-selectable sampling rates of 4, 6, or 8k Hz so that the optimum quality storage trade-offs can be selected. Onboard output amplifier and volume control are included. Adaptive delta pulse code modulation is used to reduce the size of digital recordings. Recordings are stored on system disk until needed. Stored messages can be accessed randomly under program control for playback.

The IBM PC model is priced at under \$350 and provides a cost-effective tool for anyone wanting to add high-quality voice output to an application. Protalker can be used for education and training applications; point-of-sales aids; providing a voice interface for real-time devices, as well as an interface for the visually impaired; and any other applications requiring voice output. Easy-to-use software allows the novice to create and play back voice message files. Source-code-to-assembly-language drivers are provided so OEMs and advanced users can connect Protalker to most programming languages and applications.

A telephone demonstration of the Protalker is available at (415) 858-2795. For further information, contact Speech, Ltd., 3790 El Camino Real, Suite 213, Palo Alto, CA 94306, telephone (415) 941-2490.

Circle 45



Voice Evaluation

The MV-500 evaluation unit from Micro-voice has been designed to assist users and engineers in first-time applications of digital speech. Having the built-in ability to record the user's own voice through a microphone, the MV-500 permits you to store and edit the messages and message segments without costly outside vocabulary programming. Once recorded, the messages can be played back in response to simple external switch closures or commands entered through a standard serial port. The MV-500 is a standalone unit that only requires DC power and a small speaker for immediate operation. With its built-in 64 Kbyte memory, the unit can store up to 30 seconds of words and phrases that can be easily combined, without any external controls, to make announcements of any length.

Intended only for developmental use, the evaluation unit demonstrates the major features of Microvoice's standard products but does not share other capabilities such as up and downloadable voice files, memory expansion options, or selectable audio bandwidths. However, the introductory MV-500 still provides higher voice quality and more onsite flexibility than any other digital speech output device on the market.

The MV-500 is available for \$995 from Microvoice Systems Corp., 23362 Peralta Drive, Suite 5, Laguna Hills, CA 92653, telephone (714) 859-1091.

Circle 46

Csharp Toolkit

Csharp™ is a real-time toolkit that helps C programmers increase their productivity in real-time environments. The Csharp Real-Time Toolkit™ consists of efficient functions written in the C language and distributed in source code. Csharp tools can benefit programmers in laboratory data acquisition, robotics, process control, and interactive graphics environments.

The Toolkit consists of five independent packages: Cevent, Cgraph, Cisir, C Sched, and Cstate. Cevent provides a high-level interface to external events such as switch closures and button presses. Cevent can also count and time these events. Cgraph lets programmers write efficient and portable graphics programs. With Cgraph, all parameters of a graphics system can be dynamically configured using C procedure calls. Cisir provides high-level access to machine-level features such as processor priority control and device interrupts. Cisir contains procedures written in both assembly language and C which access critical processor features. C Sched controls the real-time execution of user procedures written in C. Programmers can schedule, cancel, and synchronize the execution of C procedures. Cstate provides structured process control using state system notation. Cstate allows programmers to define multiple state systems.

Csharp tools are processor-independent and run on the entire PDP-11 family, as well as other popular 8- and 16-bit processors. Csharp tools will run under many operating systems including RT11™ and UNIX™. The toolkit is available from Systems Guild, Inc., PO Box 1085, Cambridge, MA 02142, telephone (617) 451-8479.

Circle 47

Prolog-86

Prolog-86™ allows you to learn quickly and easily how to program in Prolog. Prolog-86 include tutorials and sample programs that help you start working with the Interpreter in minutes. You should get a feel for the language and its power in the first few hours. Prolog-86 is ideal for getting a quick and thorough exposure to the language and for experimenting with it. No

programming experience is necessary, but an understanding of logic, inference, and of systematic relationships will be of great value. Although Prolog-86 can work with fairly large applications, it is not designed for use in full-scale commercial applications with tens of thousands of facts.

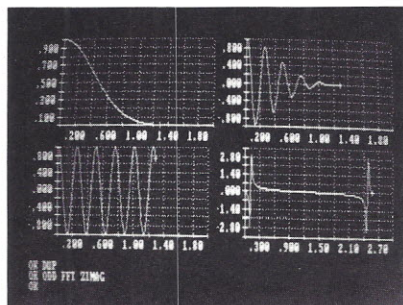
Prolog-86 requires any version of MS-DOS or CP/M-86, at least 96 Kbytes of

memory, and at least one floppy disk drive. Prolog-86 is available from \$125. The book "Programming in Prolog" by Clocksin and Mellish is a recommended but not required reference, for an extra \$20. Prolog-86 is available from Solution Systems, 335 Washington Street, Norwell, MA 02061, telephone (617) 871-5435.

Circle 48

New Products

Engineering ASYST



ASYST™ is an integrated, interactive software package for scientists, engineers, and mathematicians. ASYST combines mainframe and minicomputer speed, precision, and flexibility in programs designed for use on the IBM PC/XT. ASYST integrates three widely used scientific research functions: data acquisition, analysis, and graphics. Users can acquire, manipulate, analyze, graphically display, and print data produced by scientific instruments and experiments.

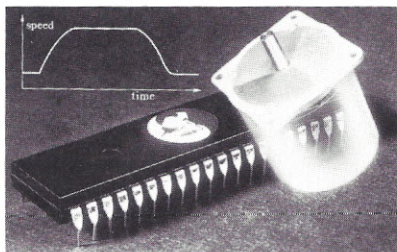
The system is composed of three modules. Module 1 provides the basic system, graphics, and statistical and math functions. Module 2 provides data analysis functions for reducing, manipulating, and analyzing data. Module 3 adds data acquisition abilities between the ASYST software and scientific and experimental equipment.

ASYST operates under standard MS-DOS and utilizes the Intel 8087 math coprocessor to achieve 80-bit internal precision and execution speeds previously found only on conventional minicomputer systems. ASYST applications can grow to the size of available memory, up to 640 Kbytes on the IBM PC. The ASYST environment uses commands written in simple conceptual terms, rather than complicated programming syntax. This makes it easy to use by both experienced and novice users. Extensive documentation includes comprehensive user manuals, a quick-reference card, and a cross-referenced index and tutorials. Each module is contained on one 5¼ in. double-sided floppy diskette. The list price for the total three-module package is under \$1800. Modules are also available separately.

For more information, contact Macmillan Software Co., 866 Third Ave., New York, NY 10022, telephone (212)702-3438.

Circle 49

Intelligent Stepper Motor Controller



The CY525 intelligent ramping stepper motor controller from Cybernetic Micro Systems is a standard 5V, 40-pin, LSI device configured to control any four-phase stepper motor. The CY525 will interface to any computer using parallel TTL input and provides numerous TTL inputs and outputs for auxiliary control. The CY525 allows sequences of high-level type commands to be stored internally in a program buffer and executed upon command. The TTL outputs

sequence the stepper drive circuits that consist of standard power transistor arrays. The CY525 features the ability to change the stepping rate while the motor is stepping and to take an unlimited number of steps in continuous run mode.

Standard features include programmability via ASCII keyboard, single 5 V power supply, 27 high-level language commands, 60 bytes of stored program capability, definable linear acceleration, ability to read position on the fly, absolute and relative position modes, two interrupt request outputs, programmable output line, programmable delay, ability to turn off phases, DOWHILE, WAIT UNTIL, and JUMP TO commands.

For more information, contact: Cybernetic Micro Systems, PO Box 3000, San Gregorio, CA 94074, telephone (415) 726-3000.

Circle 50

The DENNING RESEARCH VEHICLE

Autonomous, Intelligent and Affordable.

Designed expressly for laboratory research in robotics and robotic programming (including artificial intelligence projects), Denning Mobile Robotics Model T gives you the capability and flexibility you've been looking for at a price you can afford.

The Model T has a large and strong chassis and drive system upon which can be mounted a meaningful payload for vision, sensor and/or software research. The vehicle is just small enough to pass through standard doors, yet large enough to provide a platform of 4 square feet for custom mounting of research equipment.

The Model T is round, with three-wheel drive/three-wheel steering to simplify turning corners, approaching and passing through doors, and handling obstacles. Make no mistake, the Denning unit's versatility in research and teaching applications is impressive.

Features include:

- Synchronous three-wheel drive/three-wheel steering
- Z80 Motor Control
- Z80 Ultrasonic Sensor Ring
- Built-in Object Avoidance
- Approach doors at any angle
- M68000 with 128K RAM for processing
- Extra card slots for research use
- Two payload platforms for research material

Call 617-935-4840 today for more information.

DENNING MOBILE ROBOTICS

21 Cummings Park, Woburn, MA 01801

New Products

6-Bit Video Digitizer

The 6-Bit PC-EYE™ Model 1100 video digitizer expands the PC-EYE family of video capture systems from Chorus Data Systems. The single printed circuit board plugs directly into a single expansion slot in the IBM PC/XT and compatibles. Resolution of up to 640 by 512 pixels with 6 bits (64 levels of gray scale) per pixel is standard. Any RS-170 or RS-330 standard video source (such as common closed circuit TV cameras) or decoded NTSC inputs from VCRs or home video cameras can be connected to the four video input channels. When captured, the image can be displayed directly in a user-specified area on the PC monitor.

Capture speed is determined primarily by the resolution selected (from 320 by 200 to 512 by 512 pixels) and the number of bits per pixel (from 1 to 6). When used with compatible graphics adaptor cards supporting DMA or if transferring to PC main memory, transfer speed will typically range from 8 frames per second to 2 frames per second. Programmed input/output is also supported for use with graphics adaptor cards not supporting DMA, and for special applications.

Software support comes in two forms. A simple menu-driven environment is provided



to familiarize the user with PC-EYE features including white/black level adjustment, picture cropping, storage, printing, and acquisition modes. A second software support level provides utility subroutine libraries that can be called from C, BASICA, or assembly language. Applications software available separately from Chorus and third party vendors includes text/image merge for word processing applications, graphic arts/slide presentation, Halo™ graphics library support, telecommunications, and inspection/measurement.

The PC-EYE Model 1100 is priced at \$610. For more information, contact: Bruce Monk, VP Marketing, Chorus Data Systems, 6 Continental Blvd., PO Box 370, Merrimack, NH 03054, telephone (603) 424-2900.

Circle 51

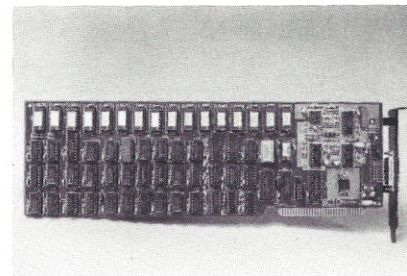


Image Digitizer

The FAX640™ Image Digitizer from Scion Corp. is a single-card digitizer for the IBM PC/XT. This single card will digitize one frame of a monochrome RS170 video image to a 16 greyscale 640 by 408 or 544 by 480 pixel resolution image—all in 1/30 second. The image memory and all-refresh logic are contained on the board. The FAX640 occupies no host memory and can maintain the grabbed image indefinitely. The digitized image can be transferred in full or part over the IBM PC data bus to Scion's PC640™ for storage, modification, or display. Image transfer takes about a second.

For more information, contact Scion's Sales Department at 12310 Pinecrest Rd, Reston, VA 22091, telephone (703) 476-6100.

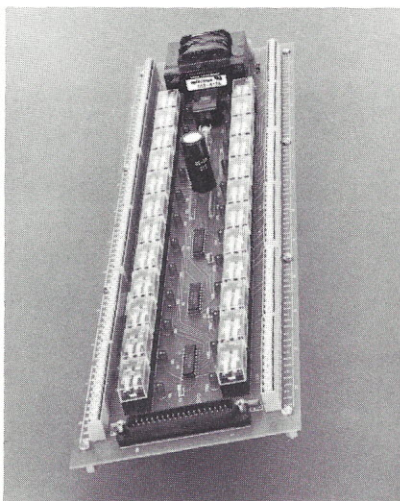
Circle 52

24 Channel DPDT Relay Output Board

MetraByte has introduced a 24-Channel DPDT Relay Output Board, Model ERB-24, that offers the programmer 24 electromechanical, double-pole, double-throw relays for efficient switching of loads by programmed control. Each relay contains two normally closed and normally open contacts for controlling up to a 3 A resistive load at 12 V RMS per contact. Designed to be operated with MetraByte's 24-Bit Parallel Digital I/O Board, Model PIO-12, it can also be operated on any digital output device providing TTL voltage levels.

The relays offer the user zero leakage output currents, compared to the solid-state relay alternative. There are 24 LEDs, one for each relay. Each lights when its associated relay is activated. The board contains its own power supply for relay driving and operates on 120 or 240 VAC, ± 15 percent at 50/60 Hz. Single piece price is \$395.

For more information, contact: Robert J.



Rogers, VP Sales & Marketing, MetraByte Corp., 254 Tosca Drive, Stoughton, MA 02072, telephone (617) 344-1990.

Circle 53

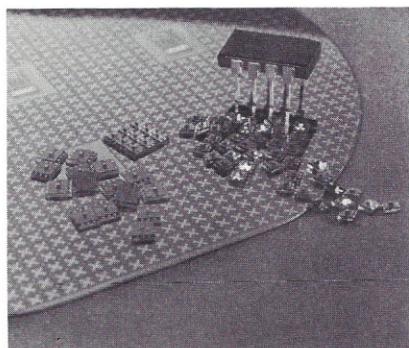
Motor Control

Control Ware has announced a new board to run two DC motors under computer control. The DC Controller Board provides independent forward, reverse and off switching of an input of up to 24 VDC at 5 A to permanent magnet motors. Four latched TTL outputs, +5 V, and ground applied to the board's DIP input connector are fully isolated from both power circuits. The controller card will work with any computer with latched TTL outputs. The card may also be hardwired to switches for initial use without a computer. Built-in protection logic and clear instructions simplify hookup and programming. A 120 day warranty and socketed ICs ease repair. The board measures 7.5 by 4 in. and is priced at \$69.00.

For more information, contact: Control Ware, P.O. Box 1467, Oak Brook, IL 60521, telephone (312) 530-4110.

Circle 54

New Products



A Sense of Touch

Transensory Devices, Inc. has introduced a new line of silicon tactile or touch sensors. The Tactile Perceptions™ line are small, low-profile devices with output signals proportional to the force applied to their surface. Available as either force switches, individual touch sensors, or multi-element tactile sensing arrays, they are designed to be mounted in end effectors or on surfaces that directly contact objects being handled in a wide variety of automation systems. Optional signal processing modules provide high-level digital signal outputs that are easily interfaced to computers or other system controllers.

An intelligent interface electronics module is available to simplify interfacing tasks. This interface contains signal processing and analog-to-digital conversion circuitry for up to 100 sensor elements along with an internal microprocessor and an RS-232 interface. Sensors or arrays can be addressed individually, as an entire array, or sector by sector. In addition, sensor temperature can also be measured and passed to the controller. The interface module also provides all necessary power and excitation to the sensors and arrays.

Two different evaluation packages are available. The TP4010K Evaluation Kit includes ten individual tactile sensors and the RS-232 interface electronics module and is priced at \$2995. A second Evaluation Kit which combines two nine-element arrays with interface electronics is \$2995. Elements, arrays, and switches are available individually at prices ranging from \$80 to \$995.

For more information, contact: Transensory Devices, Inc., 44060 Old Warm Springs Blvd., Fremont, CA 94538, telephone (415) 490-3333. Circle 55

A TOTALLY NEW GUIDE BOOK TO SENSOR AND TRANSDUCER TECHNOLOGIES

The 1984 Sensor and Transducer Directory is now available from the publishers of the monthly magazine, *Sensors: The Journal of Machine Perception*. This is the first edition of an annual directory to be published every January. It contains more than 300 listings of companies which make hundreds of components important to the engineer who must add sensors and real world interface components to microprocessor-based systems.

You'll find information organized and indexed into two sections for fast answers to your sensor and transducer technology questions. The first section is a complete master list of sensor and transducer manufacturers, corporate information and a company profile. Along with each company's listing there is a list of distributors and representatives by geographic location.

In the second section, we've provided a set of indices to the com-

pany profiles by sensor type. You'll find information on who makes nearly 100 different classes of sensors ranging from accelerometers to thermocouples, from gas pressure to humidity, from acidity to resistance. If a physical, chemical or biological quantity can be readily turned into an electronic signal that can be analyzed, you'll find a pointer to the manufacturer of such a sensor/transducer product. For each type of sensor, you'll find an alphabetical listing of company names and page numbers, giving you fast access to the suppliers of each type of sensor.

If you've ever had a tough problem finding out who makes what kind of sensor, **The 1984 Sensor and Transducer Directory** will be indispensable.

Order yours today.

ONLY \$29.95

☐ Please send _____ copies of the **1984 Sensor and Transducer Directory** at \$29.95 (includes postage and handling costs). Total Enclosed \$ _____

Overseas orders please add \$8.00 for airmail postage.

☐ Personal Check

☐ Money Order

☐ MasterCard

☐ Visa

11/84 RA

Account No. _____

Expires _____

Signature _____

Name (Please Print) _____

Company _____

Phone _____

Address _____

City _____

State _____

Zip _____

Send to **North American Technology**

174 Concord St., Peterborough, NH 03458 (603) 924-7136



Here's a robotics idea that grows with you.



Introducing Tutor™, a modular, reconfigurable robotics kit.

The best way to learn about robotics is to build a robot. And after you've built it, you'll probably think of ways to build it better. That's exactly why we designed Tutor.

The Cybot Tutor™ is actually an educational robotics kit — an environment for studying the principles of robotics in detail. It gives you hands-on experience building, programming, and using robots.

Tutor's modularity lets you begin with the basics. Controlling a stepper motor is easy. Add optical encoders for feedback. Tutor's arm gets you into linear and geometric motion control. Add the turret assembly and the Cybot Gripper™ and you're off into complex three dimensional robotics applications.

Going from one experiment to the next is easy with Tutor. For example, a 5-axis robot can be assembled in under 30 minutes!

Even the electronics are modular. Tutor's control electronics use the IEEE-696 bus (S-100). A 16-bit cpu controls Tutor while 8-bit microprocessors control the stepper motors. This lets Tutor move six stepper motors simultaneously. And extra bus boards are available for control or I/O boards that you might want to add.

An RS-232 port lets Tutor communicate with most personal computers. If you can run a serial printer, you can control Tutor.

Some of the best news is Tutor's price. You can get this industrial-grade educational robotics kit for only \$3,395.00.

Here's how to get your hands on Tutor and get into robotics, call:

800-547-4000 Dept. 610

In Oregon, or outside the USA, call:
503-684-3000 Dept. 610

To order direct, send checks or purchase orders to: Cybot, Inc.,
733 7th Ave., Kirkland, WA 98033.

Distributor and Representative inquiries are invited.

CYBOT
A Robotics Company